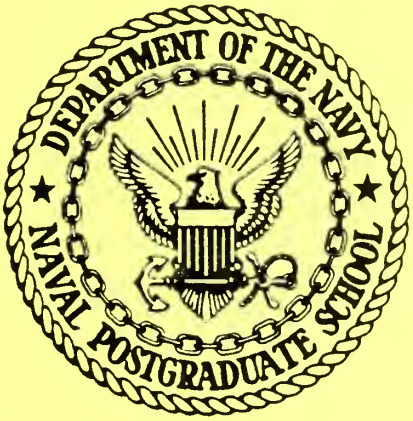
LIBRARY
TECHNICAL REPORT SECTION
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

NPS53-78-002

NAVAL POSTGRADUATE SCHOOL

Monterey, California



SMOOTH SURFACE APPROXIMATION BY A LOCAL METHOD

OF INTERPOLATION AT SCATTERED POINTS

RICHARD FRANKE

Final Report for Period January - March 1978

Approved for Public Release; distributed unlimited

Prepared for: Chief of Naval Research
Arlington, VA 22217

FEDDOCS
D 208.14/2:NPS-53-78-002

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral Tyler F. Dedman
Superintendent

Jack Borsting
Provost

The work reported herein was supported by the Foundation Research Program of the Naval Postgraduate School with funds provided by the Chief of Naval Research.

Reproduction of all or part of this report is authorized.

This report was prepared by:

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NPS53-78-002

2. GOVT ACCESSION NO.:

3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle): Smooth Surface Approximation by a Local Method of Interpolation at Scattered Points

5. TYPE OF REPORT & PERIOD COVERED: Final, Jan - Mar 1978

6. PERFORMING ORG. REPORT NUMBER:

7. AUTHOR(s): Richard Franke

8. CONTRACT OR GRANT NUMBER(s):

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Department of Mathematics
Naval Postgraduate School
Monterey, CA 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:
61152N, RR 000-01-01,
N0001478WR80023

11. CONTROLLING OFFICE NAME AND ADDRESS:
Chief of Naval Research
Arlington, VA 22217

12. REPORT DATE: April 1978

13. NUMBER OF PAGES: ~~55~~ 54

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):

15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report):
Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):

18. SUPPLEMENTARY NOTES:

19. KEY WORDS (Continue on reverse side if necessary and identify by block number):
scattered data, interpolation, surface approximation, optimal approximation, local methods

20. ABSTRACT (Continue on reverse side if necessary and identify by block number):
This report describes a computer program which constructs a surface passing through a set of data points $(x_k,y_k,f_k),k = 1,\ldots,n$. It is based on previous work of the author (JIMA 19 (1977) 471-482), but uses a somewhat different approach which takes advantage of the nature of the approximations used and incorporates experience gained in the ensuing period. The surfaces are defined for all (x,y) points and have continuous second partial derivatives.

DD FORM JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

1. The Interpolation Problem

Given the set of data points (x_k,y_k,f_k), it is desired to construct a function $F(x,y)$, such that $F(x_k,y_k) = f_k, k = 1,\ldots,n$. For large sets of data it is desirable for the method to be local, that is, the value of $F(x,y)$ depends only on the value of f_k at nearby points (x_k,y_k).

This problem is receiving a great deal of attention and discussions of it and proposed methods can be found in [1], [2], [4], [7], and [8].

2.0. The Interpolation Scheme

This is a local method, the general idea having been discussed in [4]. The basic idea is to construct local interpolants, F_ℓ, which are then weighted by functions W_ℓ having limited support to obtain the function

$$F(x,y) = \sum_\ell W_\ell(x,y)F_\ell(x,y)/\sum_\ell W_\ell(x,y) \qquad (1)$$

The details are fully discussed in the reference, but the important fact is that $F(x,y)$ will take on the value f_k at (x_k,y_k) if for each ℓ where $W_\ell(x_k,y_k) = \neq 0$, $F_\ell(x_k,y_k) = f_k$. In the referenced paper, the weight functions W_ℓ were taken to be of the form

$$W_\ell(x,y) = \begin{cases} 1 - 3(d_\ell/r_\ell)^2 + 2(d_\ell/r_\ell)^3 & , \ d_\ell \le r_\ell \\ 0 & , \ d_\ell > r_\ell \end{cases}$$

where r_ℓ is the radius of the smallest circle centered at (x_ℓ,y_ℓ) which contains a given fixed number of data points, and d_ℓ is the distance from (x,y) to (x_ℓ,y_ℓ).

This scheme, used with local interpolants, F_ℓ which were taken to be certain optimal approximations, yielded reasonably good results. However, the computational burden was rather high. This was due in part to a great deal of overlap in the regions where the W_ℓ are nonzero. In addition, the approxi-

mation is defined only on the union of the circles around each of the data points, which could cause problems.

One of the advantages of using optimal approximations is that the basis functions are generated by the (x_k,y_k) and the system is automatically nonsingular. Within some limitations, the resulting equations can easily be solved by Cholesky decomposition [5]. This opens the way for the present approach, which is to choose rectangular regions on which weight functions are non-zero, thus being able to carefully govern the amount of overlap with a resulting decrease in the necessary computations, as well as simplification of the weight functions.

2.1. Weight Functions

With these ideas in mind, we are now ready to describe the present selection of regions over which the weight functions are non-zero. These regions will be rectangles defined by the following parameters. Let n_x and n_y be given positive integers and let finite values of $\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_{n_x}$ and $\tilde{y}_1,\tilde{y}_2,\ldots,\tilde{y}_{n_y}$ be given. For convenience in this section we let $\tilde{x}_0 = \tilde{y}_0 = -\infty$ and $\tilde{x}_{n_x+1} = \tilde{y}_{n_y+1} = \infty$. For each $i = 1,\ldots,n_x$ and $j = 1,\ldots,n_y$, let R_{ij} denote the rectangle $(\tilde{x}_{i-1},\tilde{x}_{i+1})X(\tilde{y}_{j-1},\tilde{y}_{j+1})$.

Let $H_5(s) = 1 - s^3(6s^2 - 15s + 10)$, the Hermite quintic satisfying $H_5(0) = 1$, $H_5'(0) = H_5''(0) = H_5(1) = H_5'(1) = H_5''(1) = 0$. We then define functions which are piecewise quintics with continuous second derivatives which have the property that they are non-zero on two intervals and satisfy

$$V_i(x_j) = \delta_{ij}\ ,\ i = 1,\ldots,n_x\ ,\ j = 0,1,\ldots,n_x + 1$$
$$U_j(y_i) = \delta_{ji}\ ,\ j = 1,\ldots,n_y\ ,\ i = 0,1,\ldots,n_y + 1\ .$$

In particular,

$$V_1(x) = \begin{cases} 1 & , x < \tilde{x}_1 \\ H_5\left(\dfrac{x - \tilde{x}_1}{\tilde{x}_2 - \tilde{x}_1}\right) & , \tilde{x}_1 \le x < \tilde{x}_2 \\ 0 & , x \ge \tilde{x}_2 \end{cases}$$

$$V_i(x) = \begin{cases} 0 & , x < \tilde{x}_{i-1} \\ 1 - V_{i-1}(x) & , \tilde{x}_{i-1} \le x < \tilde{x}_i \\ H_5\left(\dfrac{x - \tilde{x}_i}{\tilde{x}_{i+1} - \tilde{x}_i}\right) & , \tilde{x}_i \le x < \tilde{x}_{i+1} \\ 0 & , x \ge \tilde{x}_{i+1} \end{cases}$$

for $i = 2,\ldots,n_x - 1$, and

$$V_{n_x}(x) = \begin{cases} 0 & , x < \tilde{x}_{n_x-1} \\ 1 - V_{n_x-1}(x) & \tilde{x}_{n_x-1} \le x < \tilde{x}_{n_x} \\ 1 & x \ge \tilde{x}_{n_x} . \end{cases}$$

The $U_j(Y)$ are dual. Then, if we define

$$W_{ij}(x,y) = V_i(x)U_j(Y),\ i = 1,\ldots,n_x,\ j = 1,\ldots,n_y ,$$

it is easily observed that the function $W_{ij}(x,y)$ has support $Cl(R_{ij})$ and that the functions form a partition of unity for the plane, i.e.,

$\sum_{i,j} W_{ij}(x,y) \equiv 1$ for all (x,y) .

These properties allow the construction of the interpolation function (1) to proceed easily since any point (x,y) is in at most four R_{ij} , and the denominator of (1) is always $\equiv 1$, allowing us to write

$$F(x,y) = \sum_{i,j} W_{ij}(x,y)Q_{ij}(x,y) .$$

We again emphasize that at most four terms in the sum are non-zero.

The appropriate choice of x_i and y_j as well as n_x and n_y depend on the data as well as the choice of local interpolating functions $Q_{ij}(x,y)$. For this reason we defer discussion of the selection of these grid lines until after we discuss the choice of $Q_{ij}(x,y)$.

2.2. Local Interpolation Functions

The only restriction on the local interpolation functions $Q_{ij}(x,y)$ are that they interpolate all data points in R_{ij} and that they are defined for all (x,y) in R_{ij}. Polynomials sometimes fail to satisfy these conditions. The use of optimal approximations in Sard corner spaces has been investigated [5], and for small numbers of data points, the approximations can be computed in straightforward fashion. One possible defect in such approximations is their lack of polynomial precision: even constants are not approximated exactl With only a slight complication this can be overcome, since by a theorem of Barnhill and Gregory [3], the boolean sum operator $B \oplus L$ has the interpolation properties of B and the function precision of L. Here we are thinking of Bf as the optimal approximation in $B_{\lceil 2,2 \rceil}$ while Lf is the least squares fit by a linear function.

The implemented version of the program embodies three options: (1) Use optimal approximations in $B_{\lceil 2,2 \rceil}$ as the local interpolation functions; (2) Use the least squares linear approximation instead of an interpolation functior and (3) Use the optimal approximation in $B_{\lceil 2,2 \rceil}$ boolean sum the least squares linear approximation. The second option yields a surface which in general does not interpolate the given data. The third option is achieved computationally as $(B \oplus L)f = (B + L - BL)f = Lf + B(f - Lf)$.

The use of the boolean sum has a desirable effect in that it removes much of the effect of linear transformations of the data on the overall approximatic

However, for complete consistency with respect to translation and change of the measure of distance, each rectangle

$$[\tilde{x}_{i-1},\tilde{x}_{i+1}]X[\tilde{y}_{j-1},\tilde{y}_{j+1}]$$

is transformed to the unit square for the optimal approximation. For these purposes, we take $\tilde{x}_0 = \min_k x_k$ and $\tilde{x}_{n+1} = \max_k x_k$, and the dual in y. The base point (a,b) is taken to be $(0,0)$ for all $i,j \geq 2$, while it is taken to be $(1,1)$ for $i = j = 1$, $(0,1)$ for $i \geq 2$, $j = 1$, and $(1,0)$ for $i = 1$, $j \geq 2$. This yields lines of discontinuity in the second derivatives which are nowhere interior to the support regions for the weight functions W_{ij}, thus assuring continuous second derivatives in the overall approximation.

The overall approximation is invariant with respect to linear transformations which leave the directions of the axes unchanged. Since lines of discontinuities in the third derivatives occur along horizontal and vertical lines the approximation is not invariant with respect to rotations.

The points associated with R_{ij} include all the points in the closure of R_{ij}. Because approximation by a linear function requires at least three points, a parameter MINPTS, is used to assure that at least MINPTS points are selected for each R_{ij}. If extra points are required, they are taken as the closest points in the sup norm, distance being measured after

$$[x_{i-1},x_{i+1}]X[y_{j-1},y_{j+1}]$$

has been transformed onto $[0,1]^2$. Presently MINPTS is set to three and this has been satisfactory. It is easily changed, if desired or necessary. For example, if some R_{ij} has only three colinear points associated with it, the scheme will fail under options (2) or (3). Then one must either increase the value of MINPTS or use option (1).

2.3. Selection of Grid Lines

It is desirable to have automatic selection of grid lines, that is, values of $\tilde{x}_i$ and $\tilde{y}_j$. This should be accomplished in some manner which results in rectangles R_{ij} which contain approximately equal numbers of points. For data which is poorly distributed this may not be possible. However, for somewhat uniformly distributed points the process we describe here works quite well.

The selection of the grid lines is determined by one parameter, called NPPR, for "number of points per rectangle." The grid lines are then chosen so that there will be approximately NPPR points in each rectangle, R_{ij}. If there are additional points added to certain rectangles to make up MINPTS points the average may be higher. The average is, of course, dependent on the data set.

Equal numbers of grid lines are chosen in each direction, that is $n_x = n_y$ Because we want NPPR points per rectangle, each subrectangle

$$(\tilde{x}_i,\tilde{x}_{i+1})X(\tilde{y}_j,\tilde{y}_{j+1})$$

should have $\frac{1}{4}$ NPPR points. Thus we want to choose $n_x = n_y$ so that $(n_x + 1)^2 \cdot \frac{1}{4}\,\text{NPPR} = n$, the total number of data points. Thus, we take n_x to be the nearest integer to $(4n/\text{NPPR})^{1/2} - 1$.

Grid lines, that is $\tilde{x}_i$ and $\tilde{y}_j$ values, are now determined by choosing these values so that approximately $n/(n_x + 1)$ points occur in each $(\tilde{x}_i,\tilde{x}_{i+1}]$ and each $(\tilde{y}_j,\tilde{y}_{j+1}]$. Specifically, let $\hat{x}_k$ denote the values of x_k given in nondecreasing order, then $x_i = \hat{x}_k$, where k is the integer nearest $in/(n_x + 1)$ for $i = 1,2,\ldots,n_x$. The selection in y is dual.

3.0. Implementation

The scheme is implemented in a set of subprograms, only one of which is normally referenced by the user. The hierarchy of subprograms is given in figure 1. A brief description of them, according to level, follows.

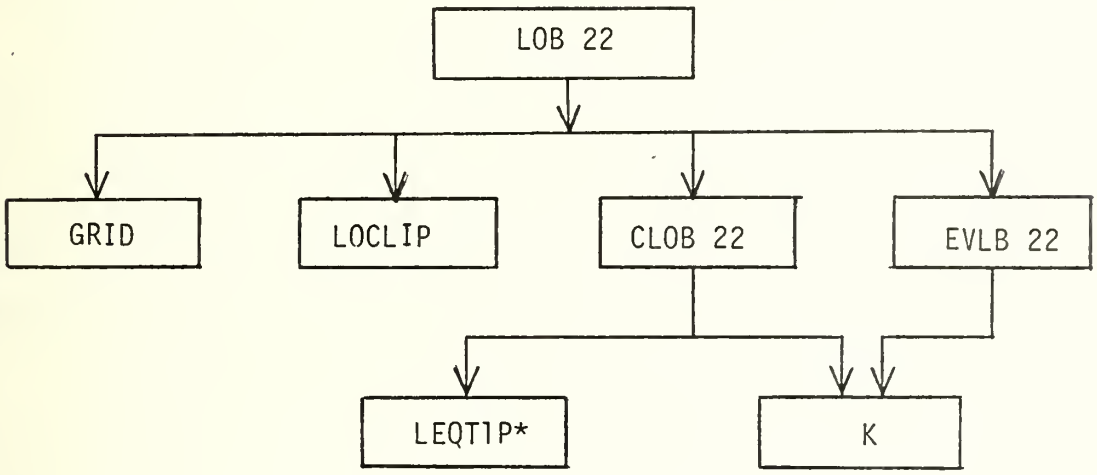


Figure 1

3.1.1. User's Program

The user's program must provide the data points, (x_k, y_k, f_k), $k = 1, \ldots, n$, as well as the points xO_i and yO_j for the grid of points at which the interpolation function is to be evaluated. In addition the user's program must provide workspace arrays, IWK and WK, and an array FO for the returned function values. The amount of storage required in the arrays IWK and WK are not known a priori, but are estimated as follows.

For IWK, approximately $4n(1 + 1/NPPR)$ locations are required. This has generally proven to be an overestimate. For WK, the storage required depends on the type of local approximation used and is approximately:

---

*This is a Cholesky decomposition equation solver in the IMSL Library, which itself references several other subroutines from the IMSL Library.

for MODE = 1, $4(n + \sqrt{n/NPPR})$;

for MODE = 2, $4(3n/NPPR + \sqrt{n/NPPR})$; and

for MODE = 3, $4(n + 3n/NPPR + \sqrt{n/NPPR})$.

As with the estimates for IWK, these estimates have proven to generally be overestimates. The precise number of storage locations required in each array are returned to the user's program by the principal subroutine.

Under the usual option, the user specifies NPPR > 0 . If the user wishes to specify the grid lines, NPPR is set to zero, and additional input in the arrays IWK and WK is necessary. In the array IWK, the user specifies n_x in IWK(1) and n_y in IWK(2). The grid lines are specified in the array WK, according to the following.

WK(1) is $\min_k x_k$,

WK(2),...,WK(n_x + 1) are the vertical grid values $\tilde{x}_1,..,\tilde{x}_{n_x}$, in increasing order,

WK(n_x + 2) is $\max_k x_k$,

WK(n_x + 3) is $\min_k y_k$,

WK(n_x + 4),...,WK(n_x + n_y + 3) are the horizontal grid values $\tilde{y}_1,...,\tilde{y}_{n_y}$ in increasing order, and

WK(n_x + n_y + 4) is $\max_k y_k$.

3.2.1. Subroutine LOB 22

This subroutine provides the interface between the user and the set of subroutines which implement the method. Generally LOB 22 sets up storage locations in the arrays IWK and WK, determines parameters required by other subroutines, and calls subroutines to (1) generate the grid (if necessary), (2) determine the interpolation points for each rectangle, (3) compute coefficients for the local interpolating functions, and finally, (4) to evaluate the interpolation function on the desired grid of points.

3.3.1. Subroutine GRID

This subroutine selects the values of $\tilde{x}_i$ and $\tilde{y}_j$ in accordance with the discussion in Section 2.3.

3.3.2. Subroutine LOCLIP

This subroutine determines the local interpolation points for each R_{ij} in accordance with the last paragraph of Section 2.2.

3.3.3. Subroutine CLOB 22

This subroutine computes the coefficients in the least squares plane (MODE = 2 or 3) and the coefficients in the optimal approximation (MODE = 1 or 3) or each of the rectangles R_{ij} . In the present implementation the IMSL Cholesky decomposition equation solver LEQT1P is used. This could be replaced at a facility where IMSL is not available, although according to IMSL policy, LEPT1P (and associated subroutines) can be used as part of this package at any facility. Because of the short single precision word length on the IBM 360/370 series computers, on which this program was implemented, the coefficients for the system of equations for the optimal approximation are generated in double precision. On computers with a longer word length, the double precision variables in this routine can be safely removed. Other double precision statements occur in EVLB 22 and function K, which must also be removed.

3.3.4. Subroutine EVLB 22

This subroutine evaluates the approximation (2) on the set of points (xO_i, yO_j) as specified by the user, and returns the values in FO. As noted above, the double precision variables in this subroutine should be removed on computers with longer word length than the IBM 360/370 series.

3.4.1. Function K

This function evaluates the representers for point evaluation functionals in $B_{[2,2]}$. For evaluation at (u,v) , base point at (a,b) , the representer, as a function of (s,t) [6] is

$$K(a,b;u,v,s,t) = g_2(a;u,s)g_2(b;v,t) \text{ , where}$$

$$g_2(a;u,s) = G_2(a;u,s) = (s-u)_+^{(3)} + 1 + (u-a)(s-a) + (u-a)(s-a)_+^{(2)} + (s-a)_+^{(3)}$$

$$\text{for } a \le u \text{ , and}$$

$$g_2(a;u,s) = G_2(-a;-u,-s) \quad \text{for } u < a \text{ .}$$

The arguments of this function are all single precision, but because of the short word length of the IBM 360/370 computers, all calculations are performed in double precision, and the returned value is double precision. On computers with longer word lengths these calculations can be done in single precision.

4.0. Examples

The method has been applied to a number of sets of data with good results Figures 3 - 5 show test surfaces and results of applying the method for each o the options for local approximations. The three surfaces are described by

(C) $F(x,y) = \tanh(y - x) + 1$,

(S) $F(x,y) = 3/2[\cos(3/5(y - 1)) + 5/4]/[1 + (\frac{x-4}{3})^2]$, and

(E) $F(x,y) = 9\{3/4 \exp(\frac{-(x-3)^2 - (y-3)^2}{4}) + \exp(-(\frac{x}{7})^2 - (\frac{y}{10}))$

$$- \frac{1}{5} \exp(-(x - 5)^2 - (y - 8)^2)$$

$$+ \frac{1}{2} \exp(\frac{-(x-8)^2 - (y-4)^2}{4})\} \text{ ,}$$

respectively. The 100 interpolation points were chosen at random within a uni square centered at (i,j) for $i,j = 1,2,\ldots,10$. The points are shown in Figure 2 as +'s, with the convex hull shown by dashed lines, while the square

$(1,10)^2$, on which the resulting interpolation functions were evaluated, is given by the solid lines. The diagonal line shows the direction toward the viewing point.

There does not appear to be a great deal of difference between the optimal approximation and the optimal approximation boolean sum least squares plane. Generally the latter option has slightly smaller errors and slightly less noticeable defects. Gross defects in the approximations can generally be traced to a lack of data in that particular part of the region.

The effect of varying the parameter NPPR is shown in Figures 6 - 14. Some general observations are possible from this set of views. Most apparent is the fact that option (2), the least squares plane fit as the local approximation does not appear to lead to very good results. In general, however, the smaller value of NPPR gives better results, visually, and usually better accuracy, too.

The choice of NPPR = 6 for options (1) and (3) appears to be a reasonable one. For surfaces with sharp gradients, as in Figure 6, it appears that localizing the behavior as much as possible with a smaller value of NPPR is the best strategy. For smooth surfaces, such as in Figure 9 and 12 it appears the opposite is true, where NPPR = 8 seems to lead to the best results.

The storage and timing results are given in Table 1. The storage refers to requirements of the two workspace arrays provided by the user. The timing is for calculation of the 1089 points generated for the plots. The program was run under the Fortran H compiler on the IBM 360 model 67 at the Naval Postgraduate School. Computation times are dependent on external factors and may vary from run to run.

5.0. Acknowledgements

During the first half of 1977 the author was a Visiting Associate Professor at the University of Utah. Interactions with Professor R. E. Barnhill and his students on the subject of surface approximation proved to be fruitful. The kernel of a number of ideas in the present scheme germinated during that time. Thanks also go to Rosemary E. Chang of Sandia Laboratories (Livermore) who first undertook to run the program on a CDC computer. Improvements in the program description and the test program were a result of those efforts.

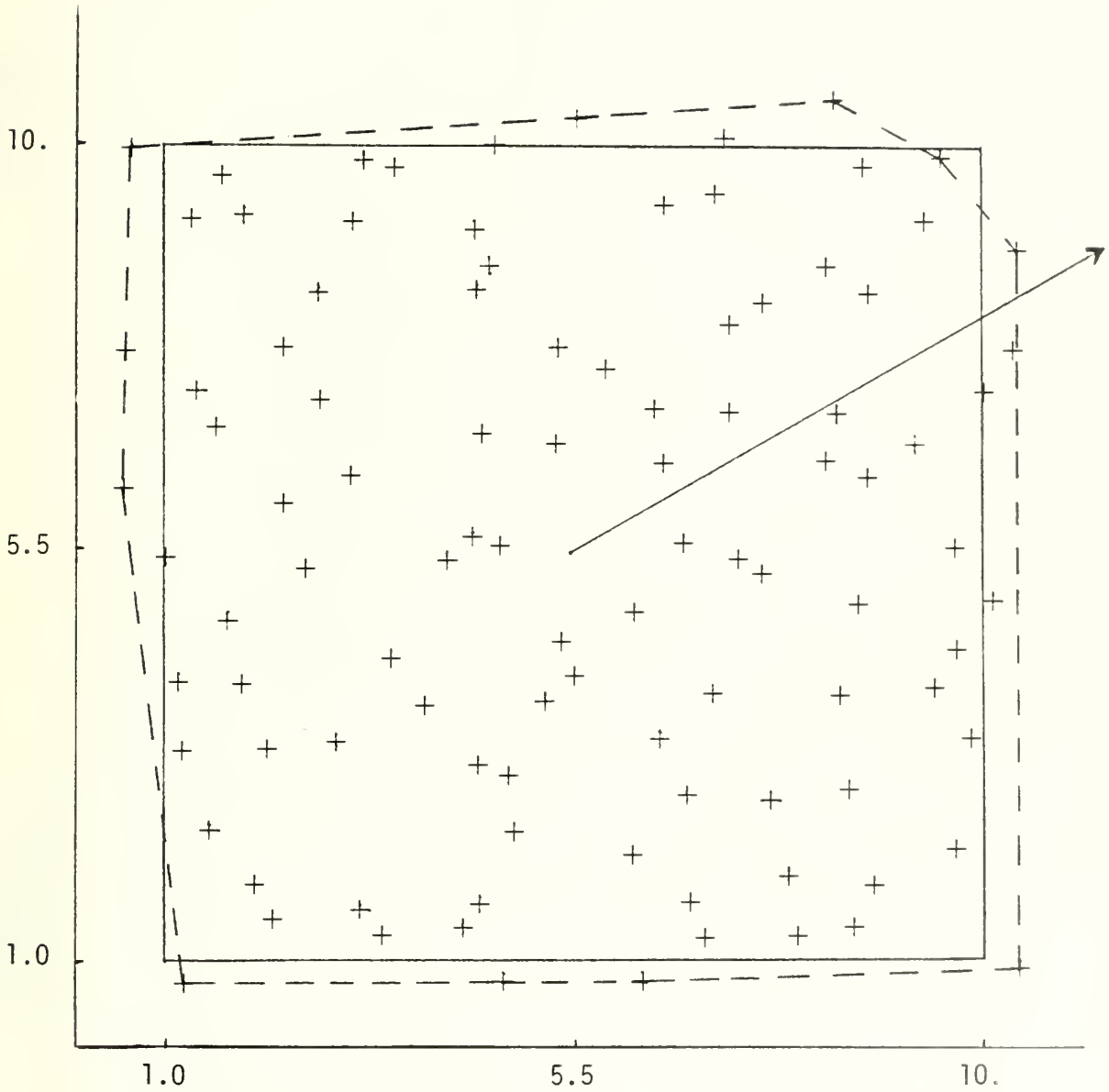
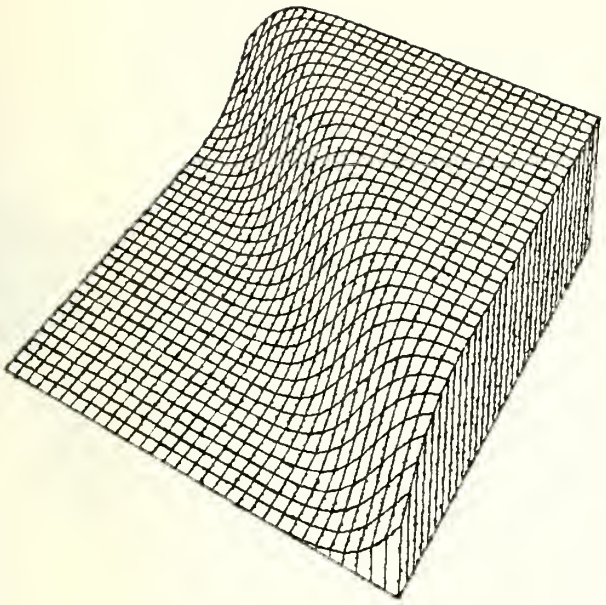


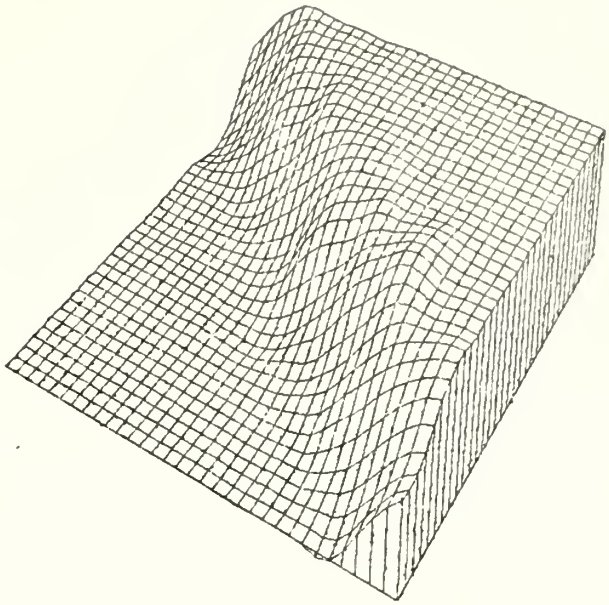
Figure 2

NPPR	MODE	NIWK	NWK	Approx. Time (sec.)
4	1	437	375	7.9
4	2	437	265	1.3
4	3	437	618	8.2
6	1	380	346	10.9
6	2	380	165	1.1
6	3	380	493	11.1
8	1	351	328	12.9
8	2	351	124	1.0
8	3	351	436	13.1

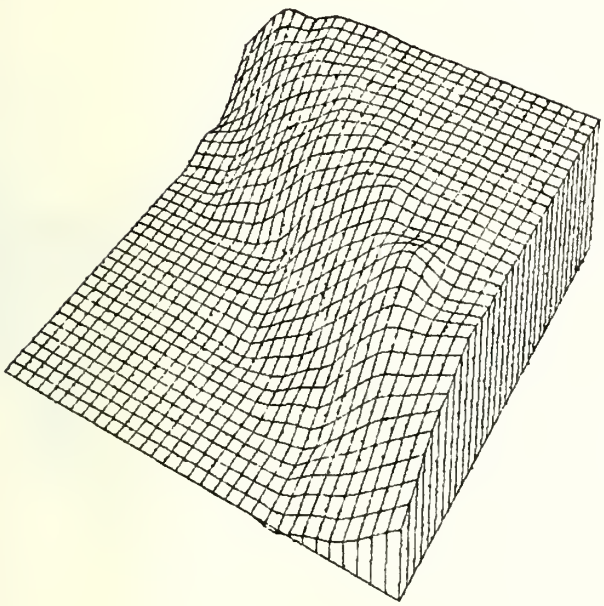
Table 1



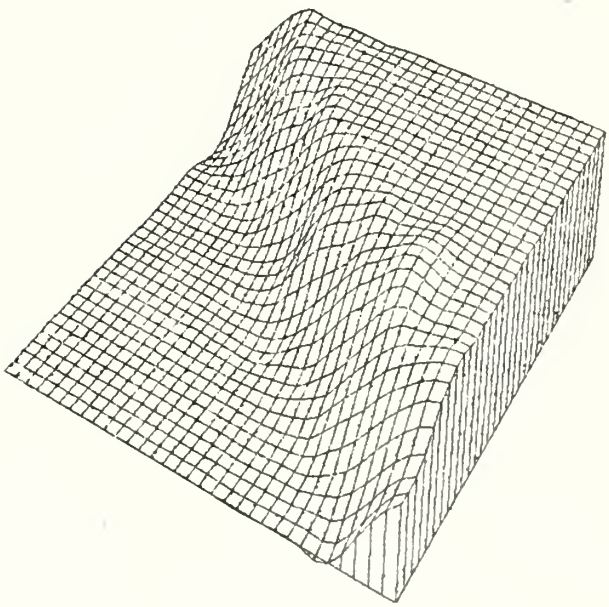
Test Surface
Cliff Function



Mode = 1 , E_{max} = .468
E_{rms} = .0263
E_{mean} = .0526

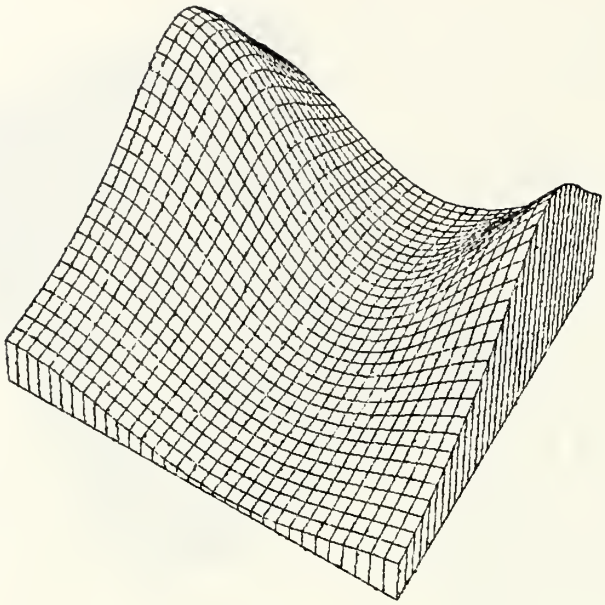


Mode = 2 , E_{max} = .283
E_{rms} = .0523
E_{mean} = .0864



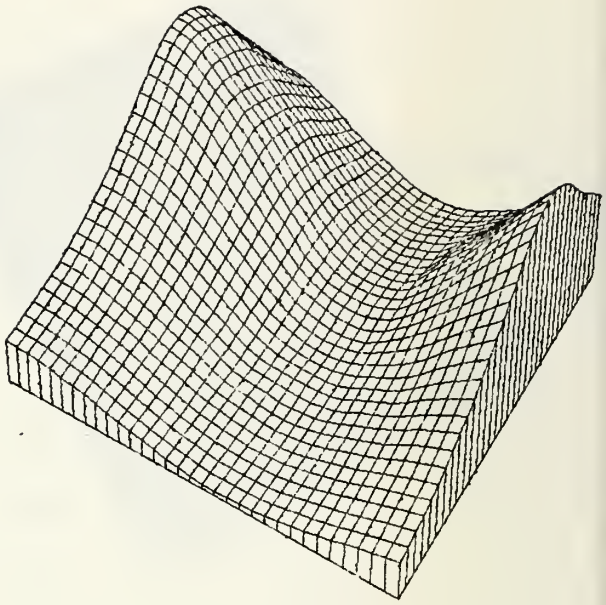
Mode = 3 , E_{max} = .466
E_{rms} = .0257
E_{mean} = .0527

Figure 3 (NPPR = 6)

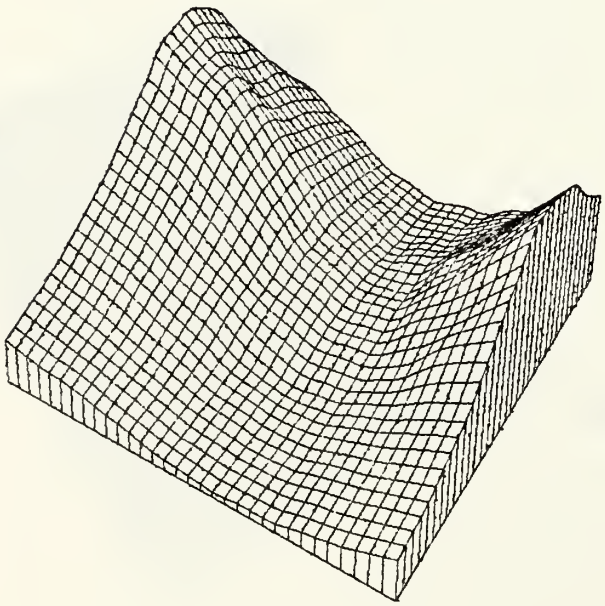


Test Surface

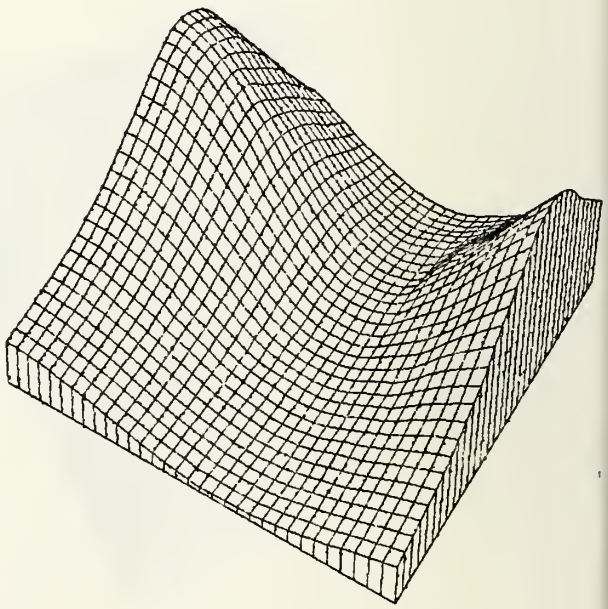
Saddle Function



Mode = 1 , E_{max} = .187
E_{rms} = .0156
E_{mean} = .0273

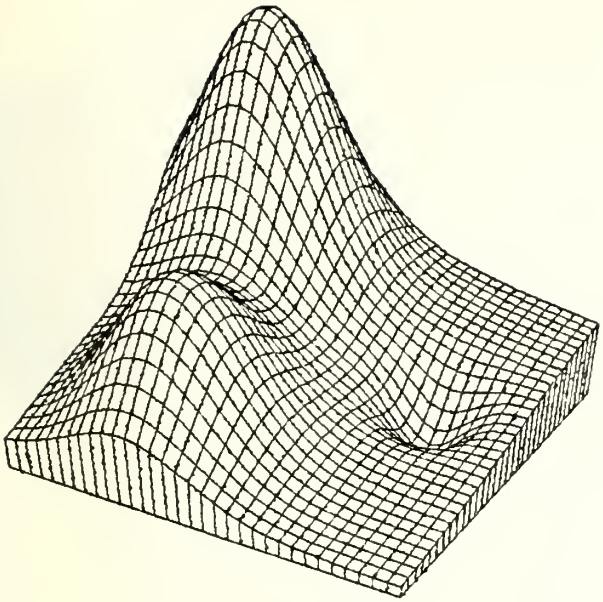


Mode = 2 , E_{max} = .389
E_{rms} = .0495
E_{mean} = .0739



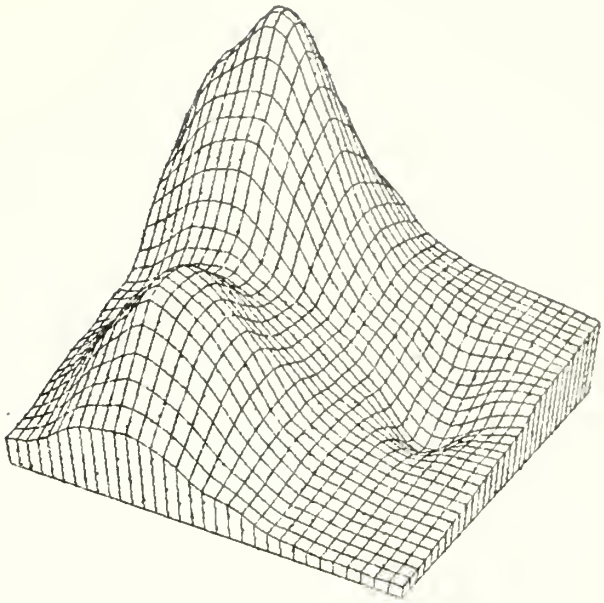
Mode = 3 , E_{max} = .178
E_{rms} = .0148
E_{mean} = .0265

Figure 4 (NPPR = 6)

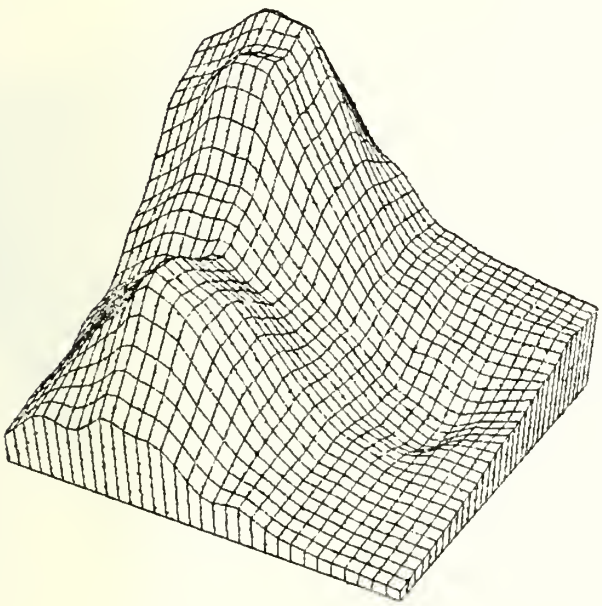


Test Surface

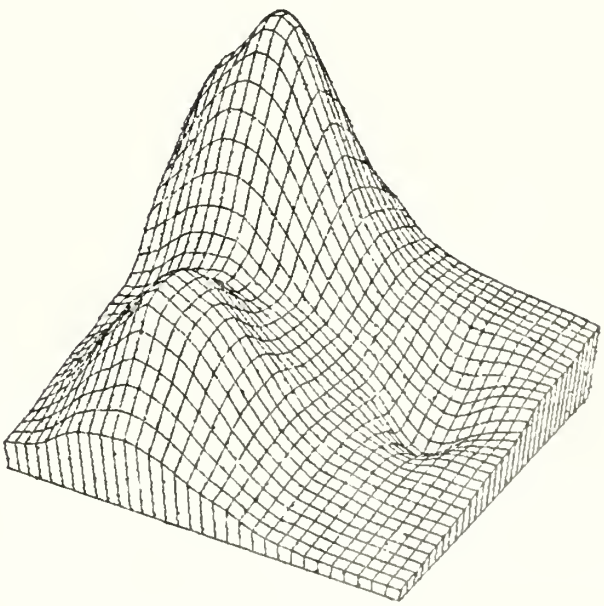
Exponentials



Mode = 1 , E_{max} = .974
E_{rms} = .0929
E_{mean} = .169

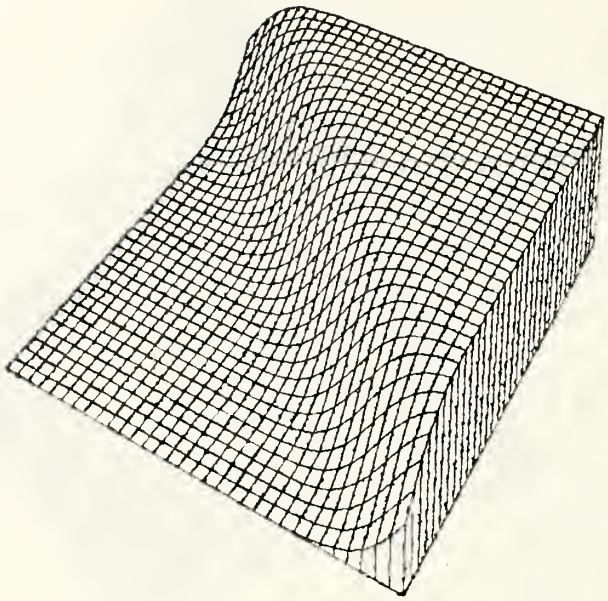


Mode = 2 , E_{max} = .216
E_{rms} = .209
E_{mean} = .366



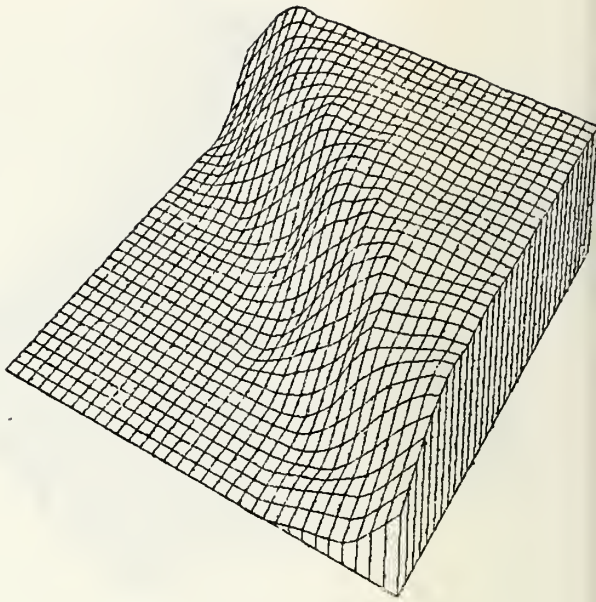
Mode = 3 , E_{max} = .827
E_{rms} = .0757
E_{mean} = .133

Figure 5 (NPPR = 6)

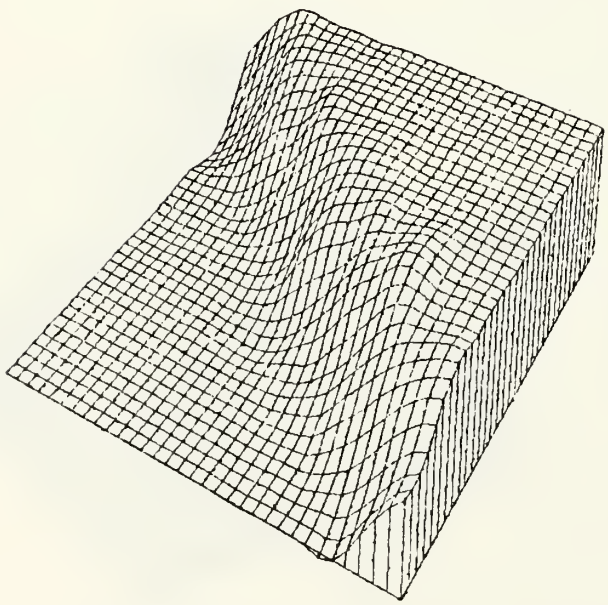


Test Surface

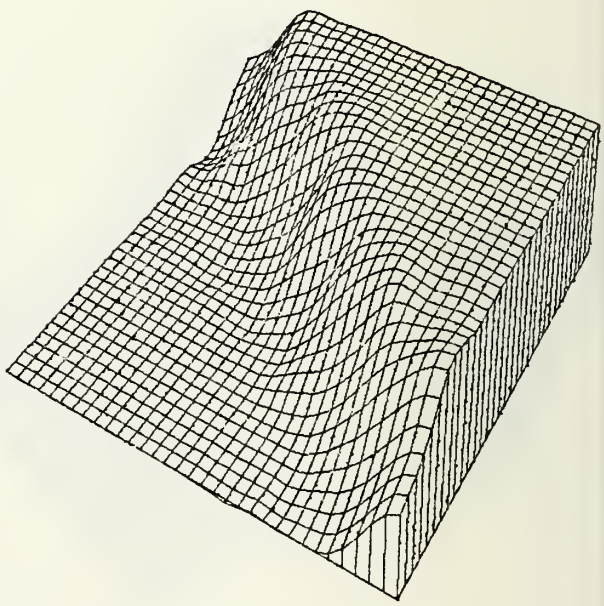
Cliff Function



NPPR = 4 , E_{max} = .265
E_{rms} = .0271
E_{mean} = .0513

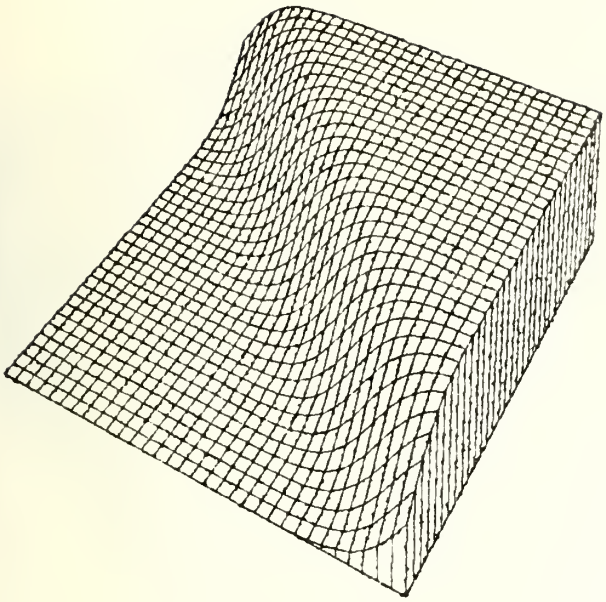


NPPR = 6 , E_{max} = .466
E_{rms} = .0257
E_{mean} = .0527



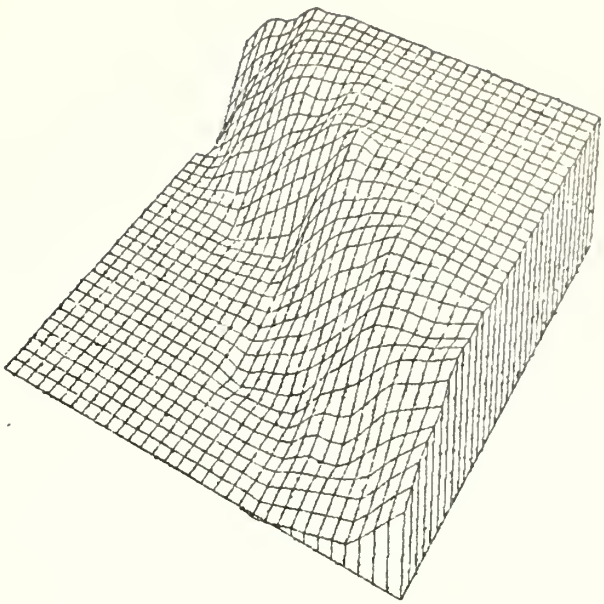
NPPR = 8 , E_{max} = .467
E_{rms} = .0308
E_{mean} = .0633

Figure 6 (Mode = 1)

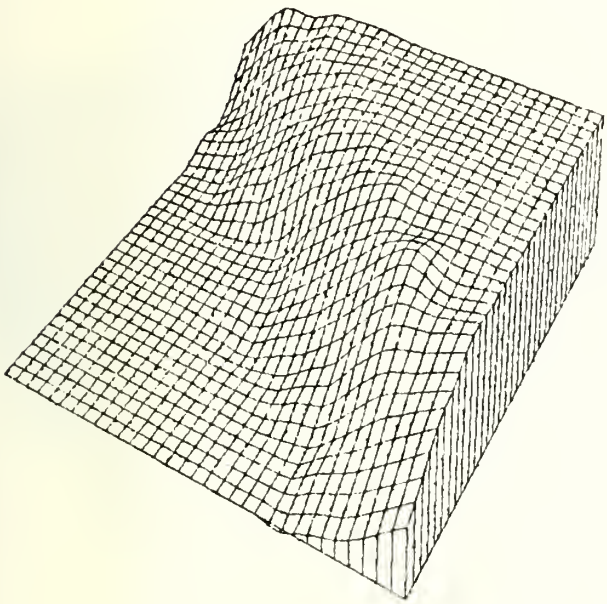


Test Surface

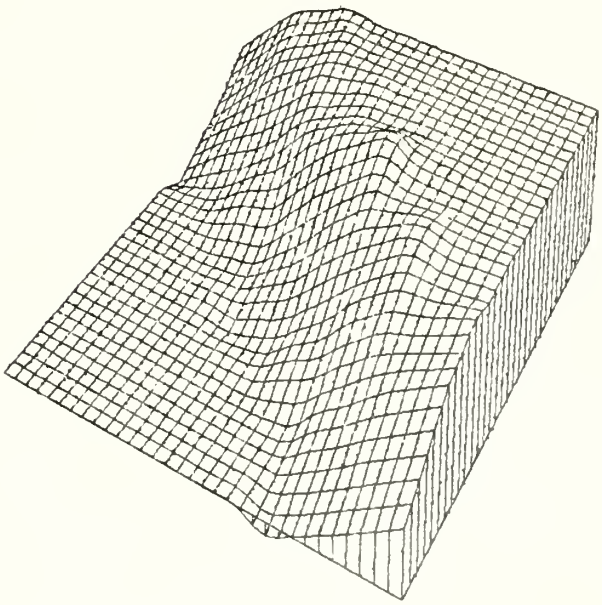
Cliff Function



NPPR = 4 , E_{max} = .336
E_{rms} = .0435
E_{mean} = .0797

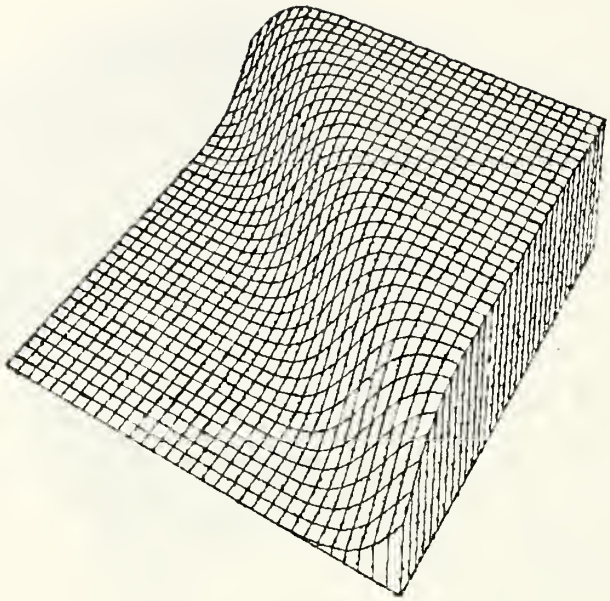


NPPR = 6 , E_{max} = .283
E_{rms} = .0523
E_{mean} = .0864

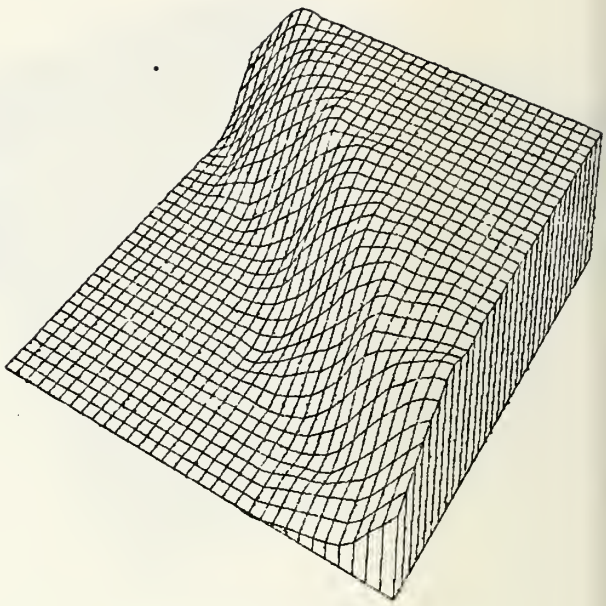


NPPR = 8 , E_{max} = .375
E_{rms} = .0692
E_{mean} = .113

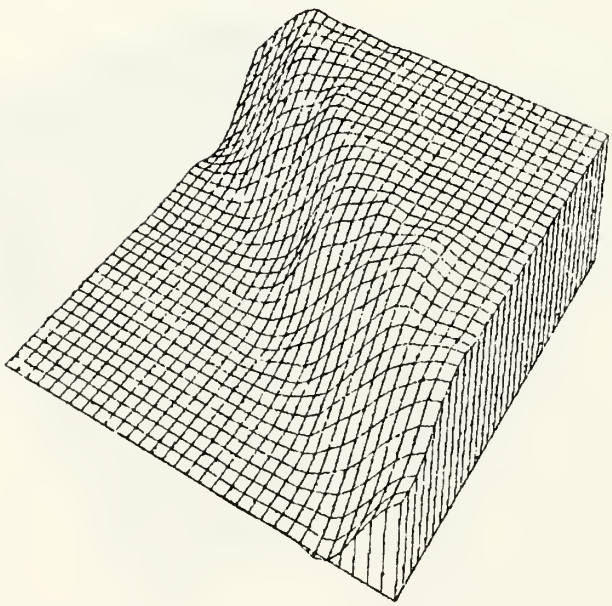
Figure 7 (Mode = 2)



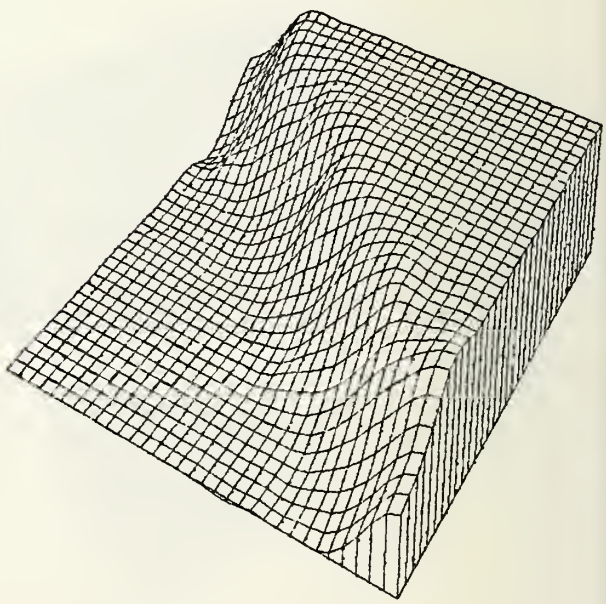
Test Surface
Cliff Function



NPPR = 4 , E_{max} = .261
E_{rms} = .0246
E_{mean} = .0500

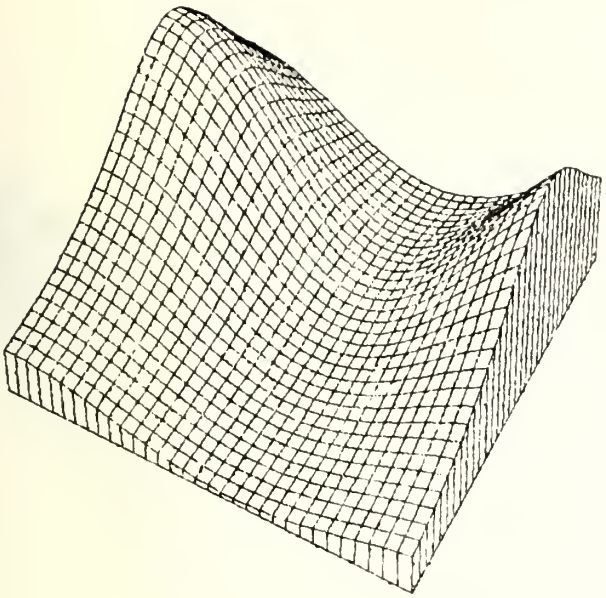


NPPR = 6, E_{max} = .468
E_{rms} = .0263
E_{mean} = .0526



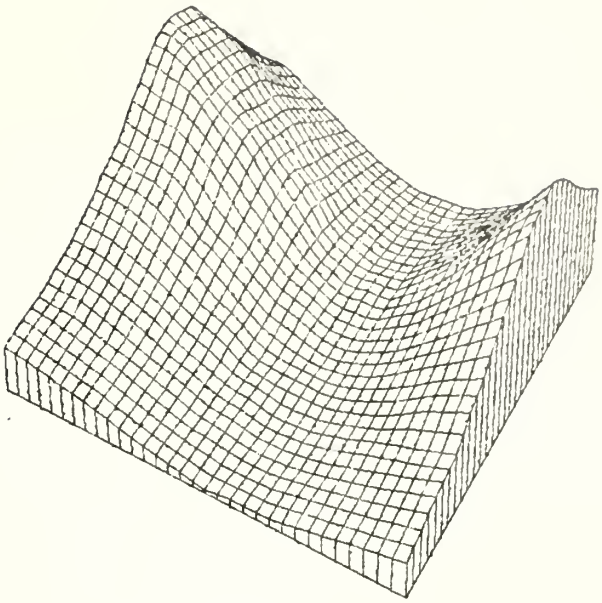
NPPR = 8 , E_{max} = .462
E_{rms} = .0304
E_{mean} = .0622

Figure 8 (Mode = 3)



Test Surface

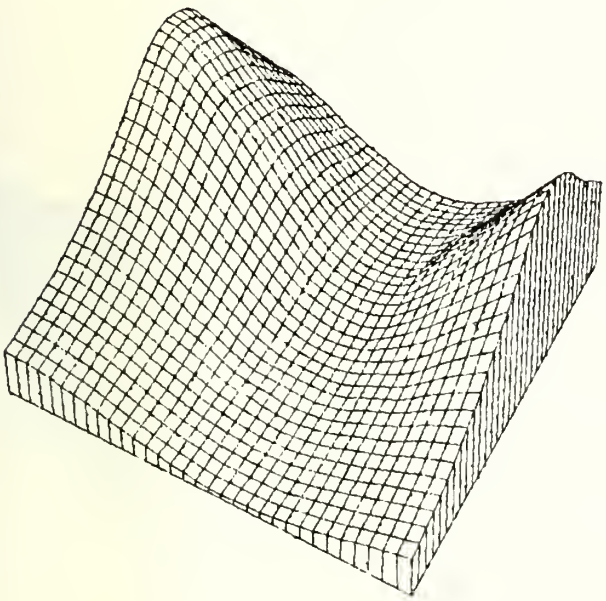
Saddle Function



NPPR = 4 , E_{max} = .208

E_{rms} = .0249

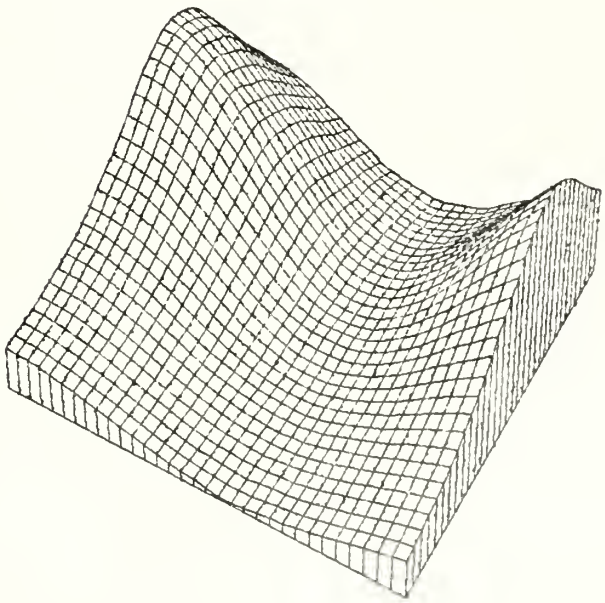
E_{mean} = .0398



NPPR = 6 , E_{max} = .187

E_{rms} = .0156

E_{mean} = .0273

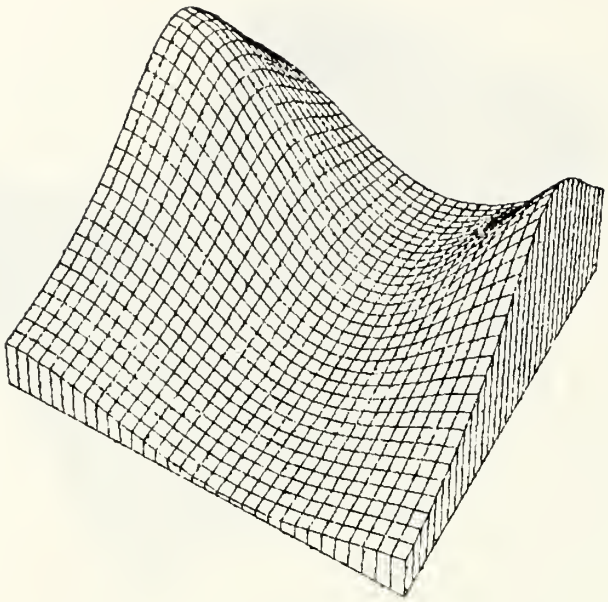


NPPR = 8 , E_{max} = .154

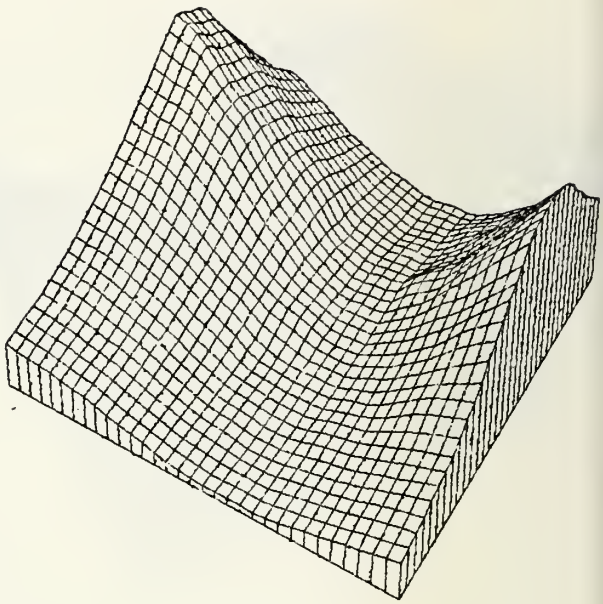
E_{rms} = .0118

E_{mean} = .0211

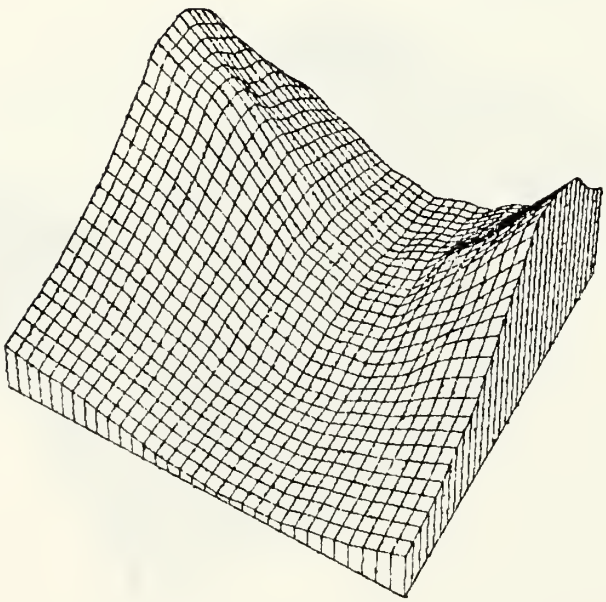
Figure 9 (Mode = 1)



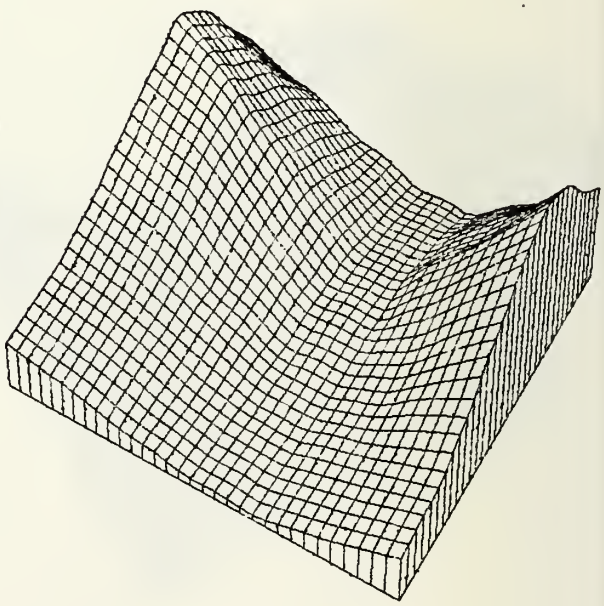
Test Surface
Saddle Function



NPPR = 4 , E_{max} = .247
E_{rms} = .0338
E_{mean} = .0487

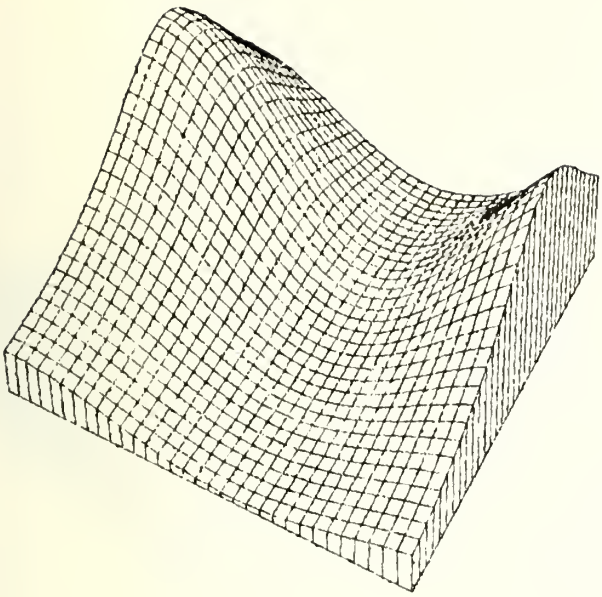


NPPR = 6 , E_{max} = .389
E_{rms} = .0495
E_{mean} = .0739



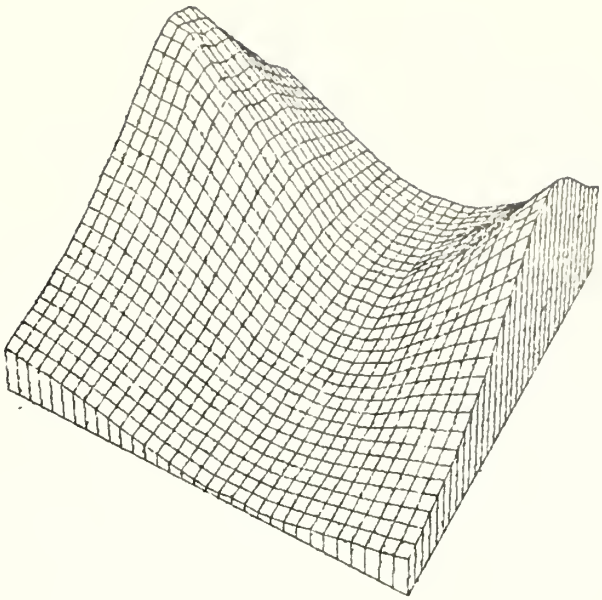
NPPR = 8 , E_{max} = .336
E_{rms} = .0565
E_{mean} = .0803

Figure 10 (Mode = 2)



Test Surface

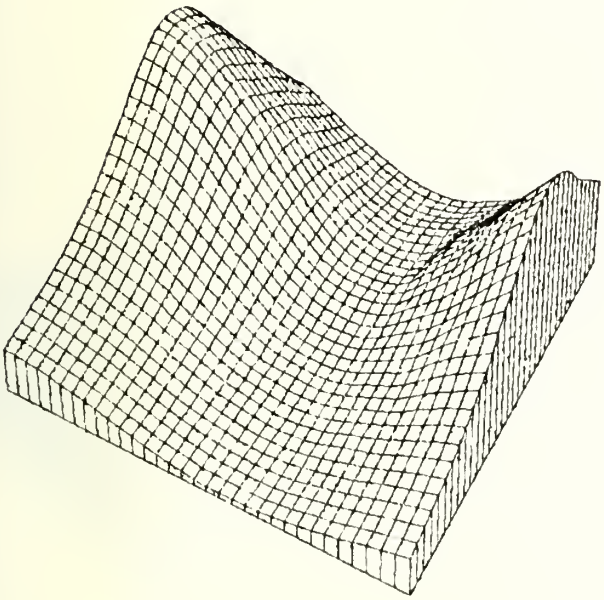
Saddle Function



NPPR = 4 , E_{max} = .244

E_{rms} = .0211

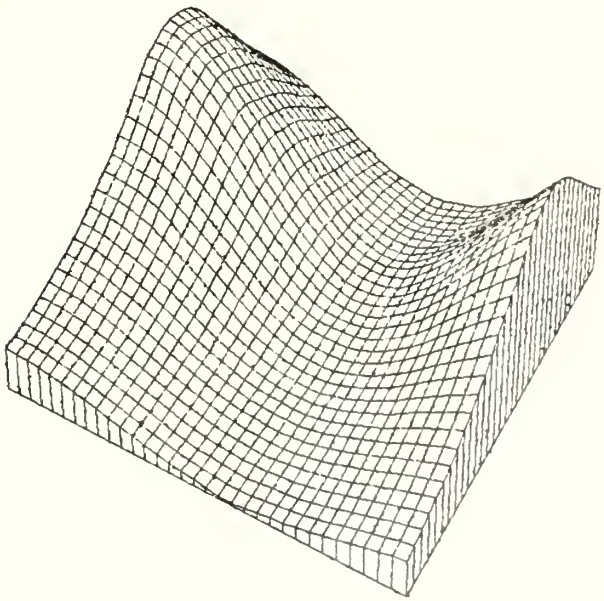
E_{mean} = .0363



NPPR = 6 , E_{max} = .178

E_{rms} = .0148

E_{mean} = .0265

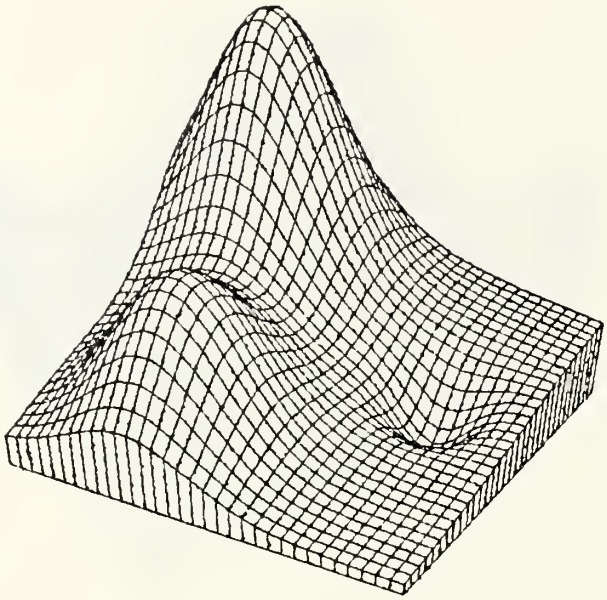


NPPR = 8 , E_{max} = .148

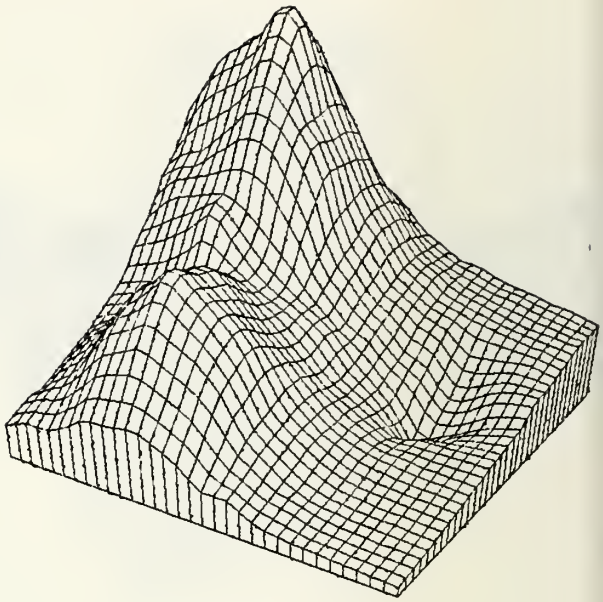
E_{rms} = .0115

E_{mean} = .0202

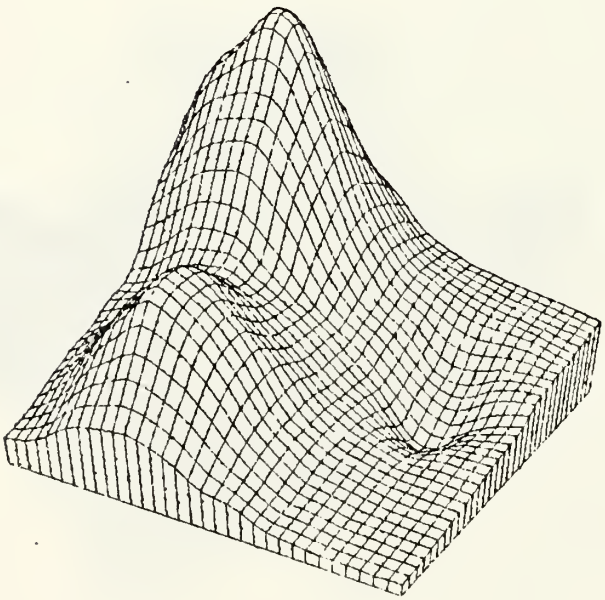
Figure 11 (Mode = 3)



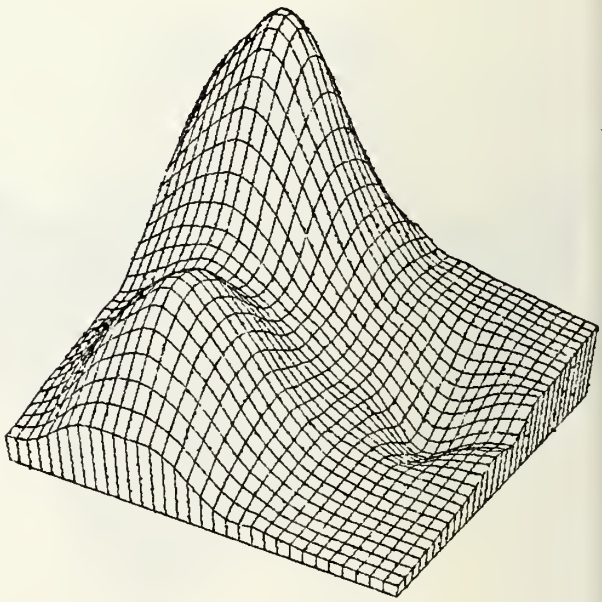
Test Surface
Exponentials



NPPR = 4 , E_{max} = 1.20
E_{rms} = .128
E_{mean} = .227

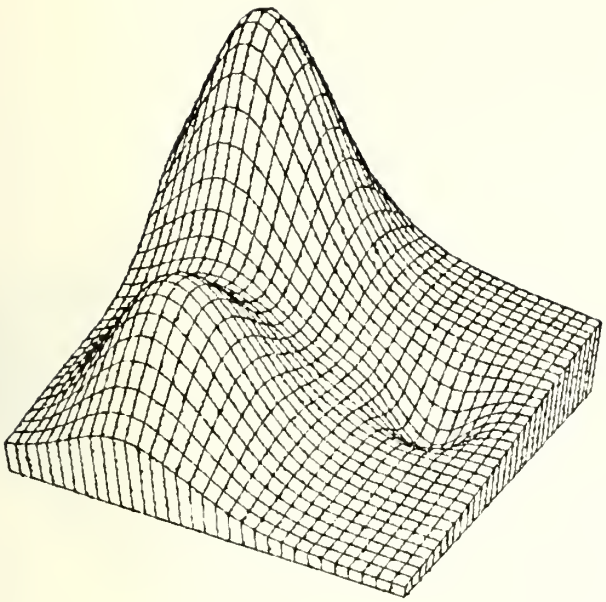


NPPR = 6 , E_{max} = .974
E_{rms} = .0929
E_{mean} = .169



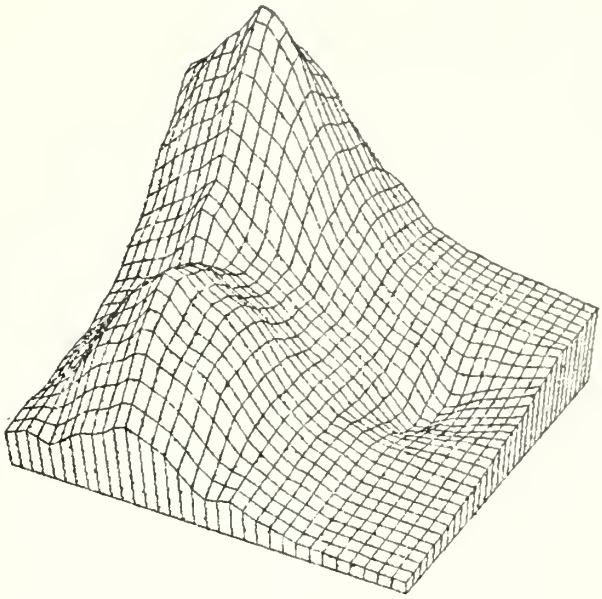
NPPR = 8 , E_{max} = .703
E_{rms} = .0779
E_{mean} = .129

Figure 12 (Mode = 1)

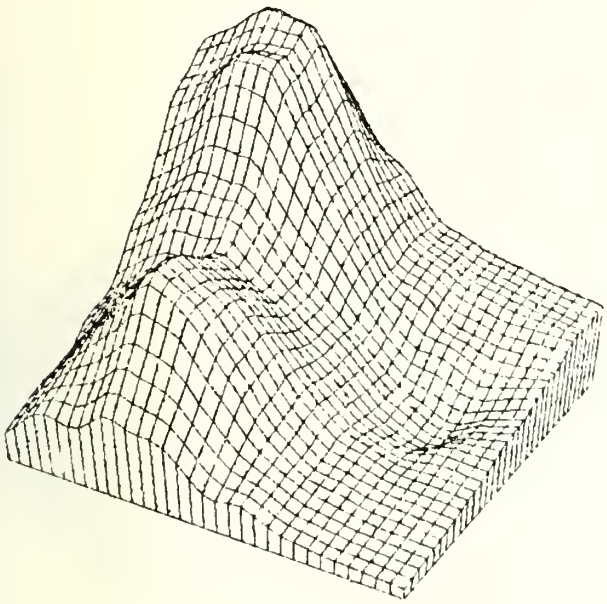


Test Surface

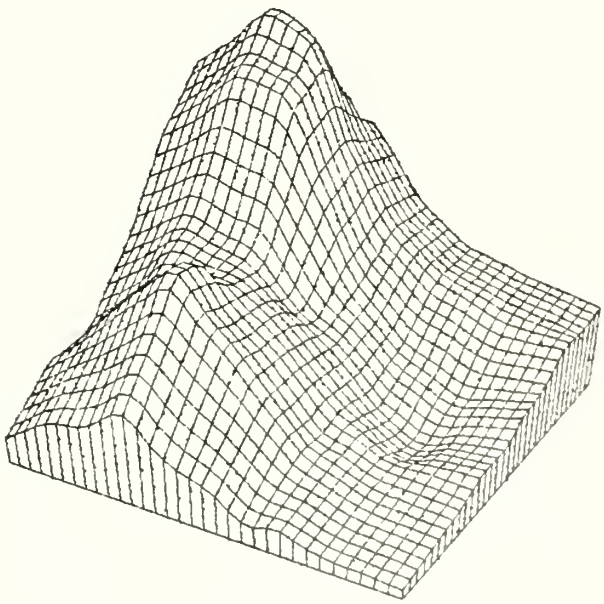
Exponentials



NPPR = 4 , E_{max} = 1.29
E_{rms} = .162
E_{mean} = .265



NPPR = 6 , E_{max} = 2.16
E_{rms} = .209
E_{mean} = .366



NPPR = 8 , E_{max} = 2.04
E_{rms} = .251
E_{mean} = .398

Figure 13 (Mode = 2)

```
      WK(5) = 0.                                                        TST00490
      WK(6) = 1.25                                                      TST00500
      WK(7) = 2.9                                                       TST00510
      WK(8) = 3.8                                                       TST00520
      WK(9) = 4.96                                                      TST00530
      NWK = 120                                                         TST00540
      NIWK = 100                                                        TST00550
      MODE = 3                                                          TST00560
      CALL LOB22 (MODE,0,NP,X,Y,F,3,XO,3,YO,IWK,NIWK,WK,NWK,FO,KER)     TST00570
C                                                                       TST00580
      DO 180 I=1,3                                                      TST00590
      DO 180 J=1,3                                                      TST00600
  180 E(I,J) = FO(I,J)-FAC(I,J,4)                                       TST00610
C                                                                       TST00620
      WRITE (NOUT,2) MODE,KER,NIWK,NWK                                  TST00630
      WRITE (NOUT,1) FO,E                                               TST00640
      STOP                                                              TST00650
C                                                                       TST00660
    1 FORMAT (/7X,15HFUNCTION VALUES,3(/3F20.6)//7X,50HDEVIATIONS (THESETST00670
     1 VALUES REPRESENT ROUNDOFF ERROR)/(3F20.6))                       TST00680
    2 FORMAT (43H0THE VALUES OF MODE, KER, NIWK, AND NWK ARE,4I5)       TST00690
      END                                                               TST00700
```

```
      SUBROUTINE LOB22 (MODE,NPPR,NPI,XI,YI,FI,NXC,XO,NYO,YO,                LOB   10
     1                  IWK,NIWK,WK,NWK,FO,KER)                              LOB   20
C                                                                            LOB   30
C     THIS SUBROUTINE SERVES AS A USER INTERFACE TO THE SET OF               LOB   40
C     SUBROUTINES THAT IMPLEMENT FRANKE'S METHOD OF SURFACE INTERPO-         LOB   50
C     LATION.  RECTANGLULAR REGIONS ARE USED WITH PRODUCT QUINTIC            LOB   60
C     HERMITE WEIGHT FUNCTIONS.  THE RECTANGLES ARE CHOSEN IN AN ATTEMPT     LOB   70
C     TO OBTAIN ABOUT NPPR POINTS IN EACH REGION.  THE SAME NUMBER OF        LOB   80
C     GRID LINES IS USED IN EACH DIRECTION.  LOCAL INTERPOLATION FUNC-       LOB   90
C     TIONS ARE EITHER OPTIMAL APPROXIMATIONS IN SARD SPACE B CORNER         LOB  100
C     2,2 OR OPTIMAL APPROXIMATIONS IN B CORNER 2,2 BOOLEAN SUM THE          LOB  110
C     LEAST SQUARES PLANE, OR FOR APPROXIMATION RATHER THAN INTERPO-         LOB  120
C     LATION, THE LEAST SQUARES PLANE.                                       LOB  130
C                                                                            LOB  140
C     THE ARGUMENTS ARE AS FOLLOWS.                                          LOB  150
C                                                                            LOB  160
C        MODE - INPUT.  INDICATES THE STATUS OF THE CALCULATION.             LOB  170
C                 = 1,  SET UP THE PROBLEM.  USE OPTIMAL APPROXIMATIONS      LOB  180
C                       FOR THE LOCAL APPROXIMATIONS, AND RETURN THE         LOB  190
C                       THE GRID OF INTERPOLATED POINTS INDICATED BY         LOB  200
C                       NXO, XO, NYO, YO, IN FO.                             LOB  210
C                 = 2,  SET UP THE PROBLEM.  USE THE LEAST SQUARES           LOB  220
C                       PLANE FOR THE LOCAL APPROXIMATION, AND RETURN        LOB  230
C                       THE GRID OF INTERPOLATED POINTS INDICATED BY         LOB  240
C                       NXO, XO, NYO, YO IN FO.                              LOB  250
C                 = 3,  SET UP THE PROBLEM.  USE THE OPTIMAL APPROXI-        LOB  260
C                       MATIONS BOOLEAN SUM THE LEAST SQUARES PLANE FOR      LOB  270
C                       THE LOCAL APPROXIMATIONS, AND RETURN THE GRID        LOB  280
C                       OF INTERPOLATED POINTS INDICATED BY NXO, XO,         LOB  290
C                       NYO, YO IN FO.                                       LOB  300
C                 = 4,  THE PROBLEM HAS BEEN SET UP PREVIOUSLY.  CALCU-      LOB  310
C                       LATE THE GRID OF INTERPOLATED POINTS INDICATED       LOB  320
C                       BY NXO, XO, NYO, YO IN FO.  THE PROGRAM ASSUMES      LOB  330
C                       THAT ALL ARGUMENTS EXCEPT NXO, XO, NYO, YO, FO,      LOB  340
C                       AND MODE ARE UNCHANGED FROM THE PREVIOUS CALL.       LOB  350
C        NPPR - INPUT.  DESIRED AVERAGE NUMBER OF POINTS PER REGION.         LOB  360
C                       THE SUGGESTED VALUE IS SIX.  SHOULD BE AT LEAST      LOB  370
C                       FOUR.  VALUES LARGER THAN TEN ARE NOT ADVISED.       LOB  380
C                       IF THE USER WISHES TO SPECIFY HIS OWN GRID LINES     LOB  390
C                       X TILDA AND Y TILDA, HE MAY DO SO BY SETTING         LOB  400
C                       NPPR = 0 AND SETTING NECESSARY VALUES IN THE         LOB  410
C                       ARRAYS IWK AND WK.                                   LOB  420
C        NPI  - INPUT.  NUMBER OF INPUT DATA POINTS.                         LOB  430
C        XI   -                                                              LOB  440
C        YI   - INPUT.  THE DATA POINTS (XI,YI,FI), I=1,...,NPI.             LOB  450
C        FI   -                                                              LOB  460
C        NXO  - INPUT.  THE NUMBER OF XO VALUES AT WHICH THE INTERP-         LOB  470
C                       OLATION FUNCTION IS TO BE CALCULATED.                LOB  480
```

```
C     XO   - INPUT.  THE VALUES OF X AT WHICH THE INTERPOLATION       LOB  490
C                    FUNCTION IS TO BE CALCULATED.                    LOB  500
C     NYO  - INPUT.  THE NUMBER OF YO VALUES AT WHICH THE INTERP-     LOB  510
C                    OLATION FUNCTION IS TO BE CALCULATED.            LOB  520
C     YO   - INPUT.  THE VALUES OF Y AT WHICH THE INTERPOLATION       LOB  530
C                    FUNCTION IS TO BE CALCULATED.                    LOB  540
C     IWK  - INPUT AND OUTPUT.  THIS ARRAY IS OUTPUT WHEN MODE = 1, 2LOB  550
C                    OR 3, AND IS INPUT WHEN MODE = 4.  THIS MUST BE  LOB  560
C                    AN ARRAY DIMENSIONED APPROXIMATELY 4*NPI*        LOB  570
C                    (1 + 1/NPPR).  FOR NPPR = 6, THIS MEANS ABOUT    LOB  580
C                    5*NPI.  WHEN NPPR IS INPUT AS ZERO THE USER MUSTLOB  590
C                    SPECIFY THE NUMBER OF VERTICAL GRID LINES (THE   LOB  600
C                    NUMBER OF X TILDA VALUES) IN IWK(1) AND THE      LOB  610
C                    NUMBER OF HORIZONTAL GRID LINES (THE NUMBER OF   LOB  620
C                    Y TILDA VALUES) IN IWK(2).                       LOB  630
C     NIWK - INPUT AND OUTPUT.  WHEN MODE = 1, 2, OR 3 THIS MUST BE   LOB  640
C                    SET TO THE NUMBER OF LOCATIONS RESERVED FOR THE  LOB  650
C                    ARRAY IWK.  ON OUTPUT IT IS SET TO THE ACTUAL    LOB  660
C                    REQUIREMENTS FOR THE ARRAY.  IF THE ARRAY IWK    LOB  670
C                    IS NOT DIMENSIONED LARGE ENOUGH IT MAY CAUSE     LOB  680
C                    A FAILURE SINCE THE ARRAY BOUNDS WILL BE         LOB  690
C                    EXCEEDED IN SUBROUTINE LOCLIP.                   LOB  700
C     WK   - INPUT AND OUTPUT.  THIS ARRAY IS OUTPUT WHEN MODE = 1, 2LOB  710
C                    OR 3, AND IS INPUT WHEN MODE = 4.  THIS MUST BE  LOB  720
C                    AN ARRAY DIMENSIONED APPROXIMATELY AS FOLLOWS.   LOB  730
C                      FOR MODE=1, 4*(NPI+SQRT(NPI/NPPR)).            LOB  740
C                      FOR MODE=2, 4*(3*NPI/NPPR+SQRT(NPI/NPPR)).     LOB  750
C                      FOR MODE=3, 4*(NPI+3*NPI/NPPR+SQRT(NPI/NPPR)).LOB  760
C                    FOR NPPR = 6, THIS MEANS APPROXIMATELY           LOB  770
C                      FOR MODE=1, 4*NPI + 1.6*SQRT(NPI)              LOB  780
C                      FOR MODE=2, 2*NPI + 1.6*SQRT(NPI)              LOB  790
C                      FOR MODE=3, 6*NPI + 1.6*SQRT(NPI)              LOB  800
C                    WHEN NPPR IS INPUT AS ZERO THE USER MUST SPECIFYLOB  810
C                    THE VALUES OF X TILDA AND Y TILDA AS FOLLOWS.    LOB  820
C                    WK(1) IS MIN(XI), WK(2),...,WK(NXG+1) ARE THE    LOB  830
C                    NXG (= IWK(1)) VALUES OF X TILDA IN INCREASING   LOB  840
C                    ORDER, AND WK(NXG+2) IS MAX(XI).  WK(NXG+3) IS   LOB  850
C                    MIN(YI), WK(NXG+4),...,WK(NXG+NYG+3) ARE THE NYGLOB  860
C                    (= IWK(2)) VALUES OF Y TILDA IN INCREASING ORDERLOB  870
C                    AND WK(NXG+NYG+4) IS MAX(YI).                    LOB  880
C     NWK  - INPUT AND OUTPUT.  ON ENTRY WITH MODE = 1, 2, OR 3, THISLOB  890
C                    MUST BE THE NUMBER OF LOCATIONS RESERVED FOR THELOB  900
C                    ARRAY WK.  ON OUTPUT IT IS SET TO THE ACTUAL     LOB  910
C                    REQUIREMENTS FOR THE ARRAY.                      LOB  920
C     FO   - OUTPUT.  VALUES OF THE INTERPOLATION FUNCTION AT THE     LOB  930
C                    GRID OF POINTS INDICATED BY NXO, XO, NYO, YO.    LOB  940
C                    FO IS ASSUMED TO BE DIMENSIONED (NXO,NYO) IN THELOB  950
C                    CALLING PROGRAM.                                 LOB  960
```

```
C         KER   - OUTPUT.  RETURN INDICATOR.                                  LOB  970
C                   = 0,   NORMAL RETURN.                                     LOB  980
C                   =-1,   PROBLEM HAS NOT BEEN PREVIOUSLY SET UP (LOB22      LOB  990
C                          CALLED WITH MODE = 1, 2 OR 3)                      LOB 1000
C                   = 1,   ERROR RETURN FROM CLOB22, SINGULAR MATRIX IN THE   LOB 1010
C                          LEAST SQUARES FIT.                                 LOB 1020
C                   = 2,   ERROR RETURN FROM CLOB22, SINGULAR MATRIX IN THE   LOB 1030
C                          OPTIMAL FIT.                                       LOB 1040
C                   = 3,   ERROR RETURN FROM CLOB22.  SOME RECTANGLE (I,J)    LOB 1050
C                          HAS MORE THAN THE ALLOWED NUMBER OF POINTS         LOB 1060
C                          ASSOCIATED WITH IT.                                LOB 1070
C                          SUBROUTINE CLOB22.                                 LOB 1080
C                   = 4,   PREVIOUS ERROR RETURN FROM CLOB22 HAS NOT BEEN     LOB 1090
C                          CORRECTED.                                         LOB 1100
C                   = 5,   IWK AND WK ARRAYS HAVE NOT BEEN DIMENSIONED        LOB 1110
C                          LARGE ENOUGH IN THE CALLING PROGRAM.  REDIMEN-     LOB 1120
C                          SION IWK AND WK TO AT LEAST THE SIZE INDICATED     LOB 1130
C                          BY NIWK AND NWK, RESPECTIVELY.                     LOB 1140
C                   = 6,   MODE IS OUT OF RANGE.                              LOB 1150
C                                                                             LOB 1160
      DIMENSION XI(1), YI(1), FI(1), IWK(1), WK(1), XO(1), YO(1), FO(NXOLOB 1170
     1,1)                                                                     LOB 1180
      DATA KERO/-1/                                                           LOB 1190
      IF (MODE.LT.1.OR.MODE.GT.4) GO TO 220                                   LOB 1200
      KER = 0                                                                 LOB 1210
C                                                                             LOB 1220
CC    ON INITIAL ENTRY MODE = 1, 2, OR 3, AND THE GRID LINES ARE SET UP,LOB 1230
CC    LOCAL INTERPOLATION POINTS ARE DETERMINED AND LOCAL APPROXIMATIONSLOB 1240
CC    ARE COMPUTED.                                                           LOB 1250
C                                                                             LOB 1260
      IF (MODE.EQ.4) GO TO 140                                                LOB 1270
      NWKIN = NWK                                                             LOB 1280
      NIWKIN = NIWK                                                           LOB 1290
      NXGWK = 1                                                               LOB 1300
      NPWK = 3                                                                LOB 1310
      IF (NPPR.LE.0) GO TO 100                                                LOB 1320
      NXG = SQRT(4.*FLOAT(NPI)/FLOAT(NPPR))-.5                                LOB 1330
      NYG = NXG                                                               LOB 1340
      IWK(1) = NXG                                                            LOB 1350
      IWK(2) = NYG                                                            LOB 1360
      GO TO 120                                                               LOB 1370
  100 NXG = IWK(1)                                                            LOB 1380
      NYG = IWK(2)                                                            LOB 1390
  120 IALWK = NXG+NYG+5                                                       LOB 1400
      IABWK = IALWK                                                           LOB 1410
      IF (MODE.NE.1) IABWK=IABWK+3*NXG*NYG                                    LOB 1420
      NYGWK = NXG+3                                                           LOB 1430
      MPWK = NXG*NYG+4                                                        LOB 1440
```

```
      IF (NPPR.GT.0) CALL GRID (XI,YI,NPI,NXG,WK(NXGWK),NYG,WK(NYGWK),WKLOB 1450
     1(IALWK))                                                          LOB 1460
C                                                                       LOB 1470
C     DETERMINE THE LOCAL INTERPOLATION POINTS FOR THE REGIONS.         LOB 1480
C                                                                       LOB 1490
      CALL LOCLIP (NXG,WK(NXGWK),NYG,WK(NYGWK),NPI,XI,YI,IWK(NPWK),IWK(MLOB 1500
     1PWK),WK(IALWK))                                                   LOB 1510
      NWK = IABWK-1                                                     LOB 1520
      IF (MODE.NE.2) NWK = NWK+IWK(MPWK-1)-1                            LOB 1530
      NIWK = NXG*NYG+2+IWK(MPWK-1)                                      LOB 1540
      IF (NIWK.GT.NIWKIN) GO TO 200                                     LOB 1550
      IF (NWK.GT.NWKIN) GO TO 200                                       LOB 1560
      MC = MODE                                                         LOB 1570
C                                                                       LOB 1580
C     COMPUTE THE LOCAL APPROXIMATIONS.                                 LOB 1590
C                                                                       LOB 1600
      CALL CLOB22 (MO,XI,YI,FI,NXG,WK(NXGWK),NYG,WK(NYGWK),IWK(NPWK),IWKLOB 1610
     1(MPWK),WK(IALWK),WK(IABWK),IER)                                   LOB 1620
      KERO = IER                                                        LOB 1630
      IF (IER.NE.0) GO TO 160                                           LOB 1640
  140 IF (KERO.NE.0) GO TO 180                                          LOB 1650
C                                                                       LOB 1660
C     COMPUTE THE FUNCTION VALUES ON THE DESIRED GRID OF POINTS.        LOB 1670
C                                                                       LOB 1680
      CALL EVLB22 (MO,XI,YI,NXG,WK(NXGWK),NYG,WK(NYGWK),IWK(NPWK),IWK(MPLOB 1690
     1WK),WK(IALWK),WK(IABWK),NXO,XO,NYO,YO,FO)                         LOB 1700
      RETURN                                                            LOB 1710
C                                                                       LOB 1720
C     ERROR RETURNS                                                     LOB 1730
C                                                                       LOB 1740
  160 KER = IER                                                         LOB 1750
      RETURN                                                            LOB 1760
  180 KER = 4                                                           LOB 1770
      IF (KERO.LT.0) KER = -1                                           LOB 1780
      RETURN                                                            LOB 1790
  200 KER = 5                                                           LOB 1800
      RETURN                                                            LOB 1810
  220 KER = 6                                                           LOB 1820
      RETURN                                                            LOB 1830
      END                                                               LOB 1840
```

```
      SUBROUTINE GRID (X,Y,N,NX,XG,NY,YG,T)                                 GRI   10
C     THIS SUBROUTINE PLACES A SET OF GRID LINES ON THE SET OF POINTS       GRI   20
C     (X,Y), I=1,...,N .  THIS IS DONE BY PLACING APPROXIMATELY EQUAL       GRI   30
C     NUMBERS OF THEM WITHIN VERTICAL AND HORIZONTAL LINES.                 GRI   40
C                                                                           GRI   50
C     THE ARGUMENTS ARE AS FOLLOWS.                                         GRI   60
C                                                                           GRI   70
C        X,Y - INPUT.  THE ARRAY OF (X,Y) POINTS.                           GRI   80
C        N   - INPUT.  THE NUMBER OF (X,Y) POINTS.                          GRI   90
C        NX  - INPUT.  THE DESIRED NUMBER OF VERTICAL GRID LINES.           GRI  100
C        XG  - OUTPUT.  THE COORDINATES OF THE VERTICAL GRID LINES.         GRI  110
C        NY  - INPUT.  THE DESIRED NUMBER OF HORIZONTAL GRID LINES.         GRI  120
C        YG  - OUTPUT.  THE COORDINATES OF THE HORIZONTAL GRID LINES.A      GRI  130
C        T   - WORK ARRAY OF DIMENSION AT LEAST N.                          GRI  140
C                                                                           GRI  150
      DIMENSION X(1), Y(1), XG(1), YG(1), T(1)                              GRI  160
      NM1 = N-1                                                             GRI  170
C                                                                           GRI  180
      DO 100 I=1,N                                                          GRI  190
  100 T(I) = X(I)                                                           GRI  200
C                                                                           GRI  210
      K = 1                                                                 GRI  220
      FINC = FLOAT(N)/FLOAT(NX+1)                                           GRI  230
      GO TO 220                                                             GRI  240
C                                                                           GRI  250
  120 DO 140 J=1,NX                                                         GRI  260
      K = J*FINC+.5                                                         GRI  270
  140 XG(J+1) = T(K)                                                        GRI  280
C                                                                           GRI  290
      XG(1) = T(1)                                                          GRI  300
      XG(NX+2) = T(N)                                                       GRI  310
C                                                                           GRI  320
      DO 160 I=1,N                                                          GRI  330
  160 T(I) = Y(I)                                                           GRI  340
C                                                                           GRI  350
      K = 2                                                                 GRI  360
      FINC = FLOAT(N)/FLOAT(NY+1)                                           GRI  370
      GO TO 220                                                             GRI  380
C                                                                           GRI  390
  180 DO 200 J=1,NY                                                         GRI  400
      K = J*FINC+.5                                                         GRI  410
  200 YG(J+1) = T(K)                                                        GRI  420
C                                                                           GRI  430
      YG(1) = T(1)                                                          GRI  440
      YG(NY+2) = T(N)                                                       GRI  450
      RETURN                                                                GRI  460
C                                                                           GRI  470
  220 DO 260 I=1,NM1                                                        GRI  480
```

```
      IP1 = I+1                                                          GRI  490
C                                                                        GRI  500
      DO 240 J=IP1,N                                                     GRI  510
      IF (T(I).LE.T(J)) GO TO 240                                        GRI  520
      TS = T(I)                                                          GRI  530
      T(I) = T(J)                                                        GRI  540
      T(J) = TS                                                          GRI  550
  240 CONTINUE                                                           GRI  560
C                                                                        GRI  570
  260 CONTINUE                                                           GRI  580
C                                                                        GRI  590
      GO TO (120,180), K                                                 GRI  600
      END                                                                GRI  610
```

```
      SUBROUTINE LOCLIP (NXG,XG,NYG,YG,NPI,XI,YI,NP,MP,D)                LOC   10
C                                                                        LOC   20
CCC   THIS SUBROUTINE DETERMINES THE LOCAL INTERPOLATION POINTS FOR THE  LOC   30
CCC   GRID VERSION OF FRANKE'S METHOD OF SURFACE INTERPOLATION.          LOC   40
CCC   MINPTS POINTS ARE REQUIRED FOR EACH REGION.                        LOC   50
CCC   IF FEWER THAN MINPTS POINTS ARE FOUND IN THE REGION, THE NEXT      LOC   60
CCC   CLOSEST POINTS (IN THE SUP NORM AFTER THE CURRENT RECTANGLE IS     LOC   70
CCC   TRANSFORMED ONTO (0,1)) ARE USED.  MINPTS IS SET TO 3, WHICH IS   LOC   80
CCC   THE RECOMMENDED VALUE, ALTHOUGH IT MAY BE ALTERED.                LOC   90
CCC                                                                      LOC  100
CCC   THE ARGUMENTS ARE AS FOLLOWS.                                      LOC  110
CCC                                                                      LOC  120
CCC      NXG   - INPUT.  NUMBER OF VERTICAL GRID LINES.                  LOC  130
CCC      XG    - INPUT.  THE COORDINATES OF THE VERTICAL GRID LINES.      LOC  140
CCC      NYG   - INPUT.  NUMBER OF HORIZONTAL GRID LINES.                LOC  150
CCC      YG    - INPUT.  THE COORDINATES OF THE HORIZONTAL GRID LINES.    LOC  160
CCC      NPI   - INPUT.  THE NUMBER OF DATA POINTS.                       LOC  170
CCC      XI    -                                                          LOC  180
CCC      YI    - INPUT.  THE DATA POINTS (XI,YI,FI), I=1,...,NPI.         LOC  190
CCC      FI    -                                                          LOC  200
CCC      NP    - OUTPUT.  AN ARRAY WHICH GIVES THE INITIAL SUBSCRIPT IN   LOC  210
CCC                      THE ARRAY MP AT WHICH THE SUBSCRIPTS FOR THE     LOC  220
CCC                      LOCAL INTERPOLATION POINTS ARE STORED.           LOC  230
CCC      MP    - OUTPUT.  AN ARRAY WHICH GIVES THE SUBSCRIPTS FOR THE     LOC  240
CCC                      LOCAL INTERPOLATION POINTS.                      LOC  250
CCC      D     - A WORK ARRAY OF DIMENSION AT LEAST NPI.                  LOC  260
C                                                                        LOC  270
      DIMENSION XG(1), YG(1), XI(1), YI(1), NP(1), MP(1), D(1)           LOC  280
      DATA MINPTS/3/                                                     LOC  290
      IJ = 1                                                             LOC  300
      NP(1) = 1                                                          LOC  310
      L = 0                                                              LOC  320
C                                                                        LOC  330
      DO 200 J=1,NYG                                                     LOC  340
      YGA = (YG(J+2)+YG(J))/2.                                           LOC  350
      DYG = YG(J+2)-YG(J)                                                LOC  360
C                                                                        LOC  370
      DO 180 I=1,NXG                                                     LOC  380
      XGA = (XG(I+2)+XG(I))/2.                                           LOC  390
      DXG = XG(I+2)-XG(I)                                                LOC  400
      IJ = IJ+1                                                          LOC  410
C                                                                        LOC  420
CCC   DETERMINE THE POINTS IN THE (I,J)TH RECTANGLE.                     LOC  430
CCC                                                                      LOC  440
C                                                                        LOC  450
      DO 120 NK=1,NPI                                                    LOC  460
      IF (XI(NK).GT.XG(I+2).OR.XI(NK).LT.XG(I)) GO TO 100                LOC  470
      IF (YI(NK).GT.YG(J+2).OR.YI(NK).LT.YG(J)) GO TO 100                LOC  480
```

```
      L = L+1                                                                   LOC  490
      D(NK) = 1.E10                                                             LOC  500
      MP(L) = NK                                                                LOC  510
      GO TO 120                                                                 LOC  520
  100 D(NK) = AMAX1(ABS(XI(NK)-XGA)/DXG,ABS(YI(NK)-YGA)/DYG)                    LOC  530
  120 CONTINUE                                                                  LOC  540
C                                                                               LOC  550
      NP(IJ) = L+1                                                              LOC  560
      IF (NP(IJ)-NP(IJ-1).GE.MINPTS) GO TO 180                                  LOC  570
C                                                                               LOC  580
C     ADD THE CLOSEST POINTS IF THERE ARE LESS THAN MINPTS IN THE               LOC  590
C     RECTANGLE.                                                                LOC  600
C                                                                               LOC  610
      LM = MINPTS-(NP(IJ)-NP(IJ-1))                                             LOC  620
C                                                                               LOC  630
      DO 160 II=1,LM                                                            LOC  640
      L = L+1                                                                   LOC  650
      MP(L) = 1                                                                 LOC  660
      DM = D(1)                                                                 LOC  670
C                                                                               LOC  680
      DO 140 NK=2,NPI                                                           LOC  690
      IF (D(NK).GE.DM) GO TO 140                                                LOC  700
      DM = D(NK)                                                                LOC  710
      MP(L) = NK                                                                LOC  720
  140 CONTINUE                                                                  LOC  730
C                                                                               LOC  740
      NK = MP(L)                                                                LOC  750
  160 D(NK) = 1.E10                                                             LOC  760
C                                                                               LOC  770
      NP(IJ) = L+1                                                              LOC  780
  180 CONTINUE                                                                  LOC  790
C                                                                               LOC  800
  200 CONTINUE                                                                  LOC  810
C                                                                               LOC  820
      RETURN                                                                    LOC  830
      END                                                                       LOC  840
```

```
      SUBROUTINE CLOB22 (MODL,XI,YI,FI,NXG,XG,NYG,YG,NP,MP,AL,AB,IER)   CLO   10
C                                                                       CLO   20
C     THIS SUBROUTINE CONSTRUCTS THE LOCAL APPROXIMANTS FOR THE GRID    CLO   30
C     VERSION OF FRANKE'S METHOD.  THE LOCAL APPROXIMATIONS MAY BE      CLO   40
C     EITHER OPTIMAL APPROXIMATIONS IN B CORNER 2,2, OR OPTIMAL         CLO   50
C     APPROXIMATIONS IN B CORNER 2,2 BOOLEAN SUM THE LEAST SQUARES      CLO   60
C     PLANE, OR FOR APPROXIMATION, RATHER THAN INTERPOLATION            CLO   70
C     THE LEAST SQUARES PLANE MAY BE SPECIFIED.                         CLO   80
C                                                                       CLO   90
C     THE ARGUMENTS ARE AS FOLLOWS.                                     CLO  100
C                                                                       CLO  110
C        MODL  - INPUT.  SPECIFIES THE TYPE OF LOCAL APPROXIMATION      CLO  120
C                        DESIRED.                                       CLO  130
C                      = 1,  USE THE OPTIMAL APPROXIMATION IN B CORNER  CLO  140
C                            2,2.                                       CLO  150
C                      = 2,  USE THE LEAST SQUARES PLANE.               CLO  160
C                      = 3,  USE THE OPTIMAL APPROXIMATION IN B CORNER  CLO  170
C                            2,2 BOOLEAN SUM THE LEAST SQUARES PLANE.   CLO  180
C        XI    -                                                        CLO  190
C        YI    - INPUT.  THE DATA POINTS (XI,YI,FI),I=1,NPI.            CLO  200
C        FI    -                                                        CLO  210
C        NXG   - INPUT.  THE NUMBER OF VERTICAL GRID LINES.             CLO  220
C        XG    - INPUT.  THE COORDINATES OF THE VERTICAL GRID LINES.     CLO  230
C        NYG   - INPUT.  THE NUMBER OF HORIZONTAL GRID LINES.           CLO  240
C        YG    - INPUT.  THE COORDINATES OF THE HORIZONTAL GRID LINES.   CLO  250
C        NP    - INPUT.  AN ARRAY WHICH GIVES THE INITIAL SUBSCRIPT IN  CLO  260
C                        THE ARRAY MP AT WHICH THE SUBSCRIPTS FOR THE   CLO  270
C                        LOCAL INTERPOLATION POINTS ARE STORED.         CLO  280
C        MP    - INPUT.  AN ARRAY WHICH GIVES THE SUBSCRIPTS FOR THE    CLO  290
C                        LOCAL INTERPOLATION POINTS.                    CLO  300
C        AL    - INPUT.  THE COEFFICIENTS FOR THE LINEAR LEAST SQUARES  CLO  310
C                        FIT, WHEN MODL = 2 OR 3.                       CLO  320
C        AB    - INPUT.  THE COEFFICIENTS FOR THE OPTIMAL APPROXIMATION CLO  330
C                        WHEN MODL = 1 OR 3.                            CLO  340
C        IER   - OUTPUT.  RETURN INDICATOR.                             CLO  350
C                  = 0,  NORMAL RETURN.                                 CLO  360
C                  = 1,  SINGULAR MATRIX HAS BEEN DETECTED IN THE       CLO  370
C                        LEAST SQUARES FIT.                             CLO  380
C                  = 2,  SINGULAR MATRIX HAS BEEN DETECTED IN THE       CLO  390
C                        OPTIMAL FIT.                                   CLO  400
C                             IN CASE OF A SINGULAR MATRIX, THE GRID    CLO  410
C                             VALUE (I,J) AND THE DATA POINTS ASSOCIATED CLO  420
C                             WITH THAT RECTANGLE ARE PRINTED.          CLO  430
C                  = 3,  THE NUMBER OF POINTS ASSOCIATED WITH SOME      CLO  440
C                        RECTANGLE  I,J  IS BIGGER THAN PRESENTLY PERMIT-CLO  450
C                        TED.  THE ARRAY C MUST BE DIMENSIONED          CLO  460
C                        NC*(NC+3)/2.                                   CLO  470
C                                                                       CLO  480
```

```
C                                                                       CLO  490
CC     BECAUSE OF THE SHORT WORD LENGTH OF THE IBM 360/370 COMPUTERS    CLO  500
CC     CERTAIN VARIABLES ARE DECLARED AS DOUBLE PRECISION.  THIS STATE- CLO  510
CC     MENT MAY BE SAFELY REPLACED WITH THE STATEMENT  ' REAL K ' WHEN  CLO  520
CC     THIS PROGRAM IS USED ON COMPUTERS WITH LONGER WORD LENGTHS.      CLO  530
C                                                                       CLO  540
      DOUBLE PRECISION K,D1,D2,C                                        CLO  550
      DIMENSION XI(1), YI(1), FI(1), NP(1), MP(1), AL(1), AB(1), XG(1), CLO  560
     1YG(1), C(230)                                                     CLO  570
      DATA NOUT,NC/6,20/                                                CLO  580
      IER = 0                                                           CLO  590
      IJ = 0                                                            CLO  600
      B = 1.                                                            CLO  610
C                                                                       CLO  620
      DO 260 J=1,NYG                                                    CLO  630
      DY = YG(J+2)-YG(J)                                                CLO  640
      A = 1.                                                            CLO  650
C                                                                       CLO  660
      DO 240 I=1,NXG                                                    CLO  670
      DX = XG(I+2)-XG(I)                                                CLO  680
      IJ = IJ+1                                                         CLO  690
      LEND = NP(IJ+1)-NP(IJ)                                            CLO  700
      IF (LEND.GT.NC) GO TO 360                                         CLO  710
      LBS = LEND*(LEND+1)/2                                             CLO  720
      IALS = (IJ-1)*3                                                   CLO  730
      IF (MODL.EQ.1) GO TO 160                                          CLO  740
C                                                                       CLO  750
CC     CALCULATE THE LEAST SQUARES PLANE                                CLO  760
CC                                                                      CLO  770
C                                                                       CLO  780
      DO 100 LL=1,9                                                     CLO  790
  100 C(LL) = 0.                                                        CLO  800
C                                                                       CLO  810
      C(1) = LEND                                                       CLO  820
C                                                                       CLO  830
      DO 120 L=1,LEND                                                   CLO  840
      MPSL = NP(IJ)+L-1                                                 CLO  850
      KL = MP(MPSL)                                                     CLO  860
      XKL = XI(KL)-XG(I+1)                                              CLO  870
      YKL = YI(KL)-YG(J+1)                                              CLO  880
      C(2) = C(2)+XKL                                                   CLO  890
      C(3) = C(3)+XKL**2                                                CLO  900
      C(4) = C(4)+YKL                                                   CLO  910
      C(5) = C(5)+YKL*XKL                                               CLO  920
      C(6) = C(6)+YKL**2                                                CLO  930
      C(7) = C(7)+FI(KL)                                                CLO  940
      C(8) = C(8)+FI(KL)*XKL                                            CLO  950
  120 C(9) = C(9)+FI(KL)*YKL                                            CLO  960
```

```
C                                                                       CLO  970
      CALL LEQT1P (C,1,3,C(7),1,0,D1,D2,KER)                            CLO  980
      IF (KER.NE.0) GO TO 280                                           CLO  990
C                                                                       CLO 1000
      DO 140 LL=1,3                                                     CLO 1010
      IAL = IALS+LL                                                     CLO 1020
  140 AL(IAL) = C(LL+6)                                                 CLO 1030
C                                                                       CLO 1040
  160 IF (MODL.EQ.2) GO TO 240                                          CLO 1050
C                                                                       CLO 1060
C     CALCULATE THE B CORNER 2,2 OPTIMAL APPROXIMATION                  CLO 1070
C                                                                       CLO 1080
      KK = 0                                                            CLO 1090
C                                                                       CLO 1100
      DO 200 LI=1,LEND                                                  CLO 1110
      MPI = NP(IJ)+LI-1                                                 CLO 1120
      KI = MP(MPI)                                                      CLO 1130
      XKI = (XI(KI)-XG(I))/DX                                           CLO 1140
      YKI = (YI(KI)-YG(J))/DY                                           CLO 1150
C                                                                       CLO 1160
      DO 180 LJ=1,LI                                                    CLO 1170
      MPJ = NP(IJ)+LJ-1                                                 CLO 1180
      KJ = MP(MPJ)                                                      CLO 1190
      XKJ = (XI(KJ)-XG(I))/DX                                           CLO 1200
      YKJ = (YI(KJ)-YG(J))/DY                                           CLO 1210
      KK = KK+1                                                         CLO 1220
  180 C(KK) = K(A,B,XKI,YKI,XKJ,YKJ)                                    CLO 1230
C                                                                       CLO 1240
      LB = LBS+LI                                                       CLO 1250
      C(LB) = FI(KI)                                                    CLO 1260
      IF (MODL.EQ.1) GO TO 200                                          CLO 1270
      XIK = XI(KI)-XG(I+1)                                              CLO 1280
      YIK = YI(KI)-YG(J+1)                                              CLO 1290
      C(LB) = C(LB)-(AL(IALS+1)+AL(IALS+2)*XIK+AL(IALS+3)*YIK)          CLO 1300
  200 CONTINUE                                                          CLO 1310
C                                                                       CLO 1320
      CALL LEQT1P (C,1,LEND,C(LBS+1),1,0,D1,D2,KER)                     CLO 1330
      IF (KER.NE.0) GO TO 300                                           CLO 1340
C                                                                       CLO 1350
      DO 220 LI=1,LEND                                                  CLO 1360
      IAB = NP(IJ)+LI-1                                                 CLO 1370
      LB = LBS+LI                                                       CLO 1380
  220 AB(IAB) = C(LB)                                                   CLO 1390
C                                                                       CLO 1400
      A = 0.                                                            CLO 1410
  240 CONTINUE                                                          CLO 1420
C                                                                       CLO 1430
  260 B = 0.                                                            CLO 1440
```

```
C                                                                       CLO 1450
      RETURN                                                            CLO 1460
C                                                                       CLO 1470
C     ERROR RETURNS                                                     CLO 1480
C                                                                       CLO 1490
  280 IER = 1                                                           CLO 1500
      GO TO 320                                                         CLO 1510
  300 IER = 2                                                           CLO 1520
  320 WRITE (NOUT,1) I,J,IER                                            CLO 1530
      MPST = NP(IJ)                                                     CLO 1540
      MPSE = MPST+LEND-1                                                CLO 1550
C                                                                       CLO 1560
      DO 340 MPS=MPST,MPSE                                              CLO 1570
      I = MP(MPS)                                                       CLO 1580
  340 WRITE (NOUT,2) I,XI(I),YI(I),FI(I)                                CLO 1590
C                                                                       CLO 1600
      RETURN                                                            CLO 1610
  360 IER = 3                                                           CLO 1620
      WRITE (NOUT,3) I,J,LEND,NC                                        CLO 1630
      RETURN                                                            CLO 1640
C                                                                       CLO 1650
    1 FORMAT (46HOSINGULAR MATRIX DETECTED FOR GRID POINT I,J =,2I5,12H CLO 1660
     1,    IER = ,I2//40H THE DATA POINTS INVOLVED ARE AS FOLLOWS/)     CLO 1670
    2 FORMAT (1X,I5,3E15.6)                                             CLO 1680
    3 FORMAT (15HOTHE RECTANGLE ,I3,1H,,I3,5H HAS ,I3,27H POINTS ASSOCIACLO 1690
     1TED WITH IT /51H THE CURRENT LIMIT IMPOSED IN SUBROUTINE CLCB22 ISCLO 1700
     2 ,I3/84H THIS MAY BE MODIFIED BY REDEFINING NC AND CHANGING THE DICLO 1710
     3MENSION OF C APPROPRIATELY)                                       CLO 1720
      END                                                               CLO 1730
```

```
      SUBROUTINE EVLB22 (MODL,XI,YI,NXG,XG,NYG,YG,NP,MP,AL,AB,          EVL   10
     1                   NXO,XO,NYO,YO,FO)                              EVL   20
C                                                                       EVL   30
CC    THIS SUBROUTINE EVALUATES THE INTERPOLANT FOR THE GRID VERSION OF EVL   40
CC    FRANKE'S METHOD.  THE FUNCTION IS EVALUATED AT THE GRID OF POINTS EVL   50
CC    INDICATED BY NXO, XO, NYO, YO, AND THESE VALUES ARE RETURNED      EVL   60
CC    IN THE ARRAY FO, WHICH IS ASSUMED TO BE DIMENSIONED (NXO,NYO).    EVL   70
CC                                                                      EVL   80
CC    THE ARGUMENTS ARE AS FOLLOWS.                                     EVL   90
CC                                                                      EVL  100
CC       MODL - INPUT.  SPECIFIES THE TYPE OF LOCAL APPROXIMATION USED. EVL  110
CC                 = 1,  USED THE OPTIMAL APPROXIMATION IN B CORNER 2,2. EVL  120
CC                 = 2,  USED THE LEAST SQUARES PLANE.                  EVL  130
CC                 = 3,  USED THE OPTIMAL APPROXIMATION IN B CORNER 2,2 EVL  140
CC                       BOOLEAN SUM THE LEAST SQUARES PLANE.           EVL  150
CC       XI   -                                                         EVL  160
CC       YI   - INPUT.  THE DATA POINTS (XI,YI,FI),I=1,...,NPI.         EVL  170
CC       FI   -                                                         EVL  180
CC       NXG  - INPUT.  THE NUMBER OF VERTICAL GRID LINES.              EVL  190
CC       XG   - INPUT.  THE COORDINATES OF THE VERTICAL GRID LINES.     EVL  200
CC       NYG  - INPUT.  THE NUMBER OF HORIZONTAL GRID LINES.            EVL  210
CC       YG   - INPUT.  THE COORDINATES OF THE HORIZONTAL GRID LINES.   EVL  220
CC       NP   - INPUT.  AN ARRAY WHICH GIVES THE INITIAL SUBSCRIPT IN   EVL  230
CC                      THE ARRAY MP AT WHICH THE SUBSCRIPTS FOR THE   EVL  240
CC                      LOCAL INTERPOLATION POINTS ARE STORED.         EVL  250
CC       MP   - INPUT.  AN ARRAY WHICH GIVES THE SUBSCRIPTS FOR THE     EVL  260
CC                      LOCAL INTERPOLATION POINTS.                    EVL  270
CC       AL   - INPUT.  THE COEFFICIENTS FOR THE LEAST SQUARES PLANE,   EVL  280
CC                      WHEN MODL = 2 OR 3.                            EVL  290
CC       AB   - INPUT.  THE COEFFICIENTS FOR THE OPTIMAL APPROXIMATION  EVL  300
CC                      WHEN MODL = 1 OR 3.                            EVL  310
CC       NXO  - INPUT.  THE NUMBER OF XO VALUES AT WHICH THE INTERPO-   EVL  320
CC                      LATION FUNCTION IS TO BE CALCULATED.           EVL  330
CC       XO   - INPUT.  THE VALUES OF X AT WHICH THE INTERPOLATION      EVL  340
CC                      FUNCTION IS TO BE CALCULATED.                  EVL  350
CC       NYO  - INPUT.  THE NUMBER OF YO VALUES AT WHICH THE INTERPO-   EVL  360
CC                      LATION FUNCTION IS TO BE CALCULATED.           EVL  370
CC       YO   - INPUT.  THE VALUES OF Y AT WHICH THE INTERPOLATION      EVL  380
CC                      FUNCTION IS TO BE CALCULATED.                  EVL  390
CC       FO   - OUTPUT.  VALUES OF THE INTERPOLATION FUNCTION AT THE    EVL  400
CC                      GRID POINTS INDICATED BY NXO, XO, NYO, YO.     EVL  410
CC                      FO IS ASSUMED TO BE DIMENSIONED (NXO,NYO) IN THEEVL  420
CC                      CALLING PROGRAM.                               EVL  430
C                                                                       EVL  440
      DIMENSION XG(1), YG(1), XI(1), YI(1), NP(1), MP(1), FC(4), AL(1), EVL  450
     1AB(1), XO(1), YO(1), FO(NXO,1)                                    EVL  460
C                                                                       EVL  470
C     BECAUSE OF THE SHORT WORD LENGTH OF THE IBM 360/370 COMPUTERS     EVL  480
```

```
C       CERTAIN VARIABLE ARE DECLARED AS DOUBLE PRECISION.  THIS STATE-   EVL   490
CC      MENT MAY BE SAFELY REPLACED WITH THE STATEMENT  ' REAL K ' WHEN   EVL   500
CC      THIS PROGRAM IS USED ON COMPUTERS WITH LONGER WORD LENGTHS.       EVL   510
C                                                                         EVL   520
        DOUBLE PRECISION K                                                EVL   530
C                                                                         EVL   540
CC      ARITHMETIC STATEMENT FUNCTION FOR THE HERMITE QUINTIC.            EVL   550
C                                                                         EVL   560
        H5(S) = 1.-S**3*((6.*S-15.)*S+10.)                                EVL   570
        J = 1                                                             EVL   580
C                                                                         EVL   590
        DO 640 JO=1,NYO                                                   EVL   600
C                                                                         EVL   610
CC      DETERMINE THE LOCATION OF THE POINT YO IN TERMS OF THE SMALLEST   EVL   620
CC      VALUE OF J SUCH THAT YO(JO) IS IN SOME RECTANGLE (I,J).           EVL   630
C                                                                         EVL   640
        YV = YO(JO)                                                       EVL   650
        JJS = J+1                                                         EVL   660
        IF (YV.LT.YG(JJS)) JJS=1                                          EVL   670
C                                                                         EVL   680
        DO 100 JJ=JJS,NYG                                                 EVL   690
        IF (YV.LT.YG(JJ+1)) GO TO 120                                     EVL   700
  100 CONTINUE                                                            EVL   710
C                                                                         EVL   720
        J = NYG                                                           EVL   730
        GO TO 140                                                         EVL   740
  120 J = JJ-1                                                            EVL   750
  140 JD = 3                                                              EVL   760
        IF (J.GE.1) GO TO 160                                             EVL   770
        JD = 0                                                            EVL   780
        J = 1                                                             EVL   790
        GO TO 180                                                         EVL   800
  160 IF (J.LT.NYG) GO TO 180                                             EVL   810
        JD = 6                                                            EVL   820
  180 DY = YG(J+2)-YG(J+1)                                                EVL   830
        I = 1                                                             EVL   840
C                                                                         EVL   850
        DO 620 IO=1,NXO                                                   EVL   860
C                                                                         EVL   870
CC      DETERMINE THE LOCATION OF THE POINT XO IN TERMS OF THE SMALLEST   EVL   880
CC      VALUE OF I SUCH THAT XO(IO) IS IN THE RECTANGLE (I,J).            EVL   890
C                                                                         EVL   900
        IIS = I+1                                                         EVL   910
        XV = XO(IO)                                                       EVL   920
        IF (XV.LT.XG(IIS)) IIS=1                                          EVL   930
C                                                                         EVL   940
        DO 200 II=IIS,NXG                                                 EVL   950
        IF (XV.LT.XG(II+1)) GO TO 220                                     EVL   960
```

```
  200 CONTINUE                                                                  EVL   970
C                                                                               EVL   980
      I = NXG                                                                   EVL   990
      GO TO 240                                                                 EVL  1000
  220 I = II-1                                                                  EVL  1010
  240 ID = 2                                                                    EVL  1020
      IF (I.GE.1) GO TO 260                                                     EVL  1030
      ID = 1                                                                    EVL  1040
      I = 1                                                                     EVL  1050
      GO TO 280                                                                 EVL  1060
  260 IF (I.LT.NXG) GO TO 280                                                   EVL  1070
      ID = 3                                                                    EVL  1080
  280 DX = XG(I+2)-XG(I+1)                                                      EVL  1090
      KD = ID+JD                                                                EVL  1100
      A = 0.                                                                    EVL  1110
      IF (I.EQ.1) A = 1.                                                        EVL  1120
      B = 0.                                                                    EVL  1130
      IF (J.EQ.1) B = 1.                                                        EVL  1140
      GO TO (300,360,300,440,520,440,300,360,300), KD                           EVL  1150
C                                                                               EVL  1160
C     THIS IS FOR (XO(IO),YO(JO)) POINTS IN A SINGLE RECTANGLE (I,J)            EVL  1170
C                                                                               EVL  1180
  300 FV = 0.                                                                   EVL  1190
      IJ = (J-1)*NXG+I                                                          EVL  1200
      IAL = 3*IJ-2                                                              EVL  1210
      IF (MODL.EQ.2) GO TO 340                                                  EVL  1220
      LMAX = NP(IJ+1)-NP(IJ)                                                    EVL  1230
      DXA = XG(I+2)-XG(I)                                                       EVL  1240
      DYA = YG(J+2)-YG(J)                                                       EVL  1250
      XVD = (XV-XG(I))/DXA                                                      EVL  1260
      YVD = (YV-YG(J))/DYA                                                      EVL  1270
C                                                                               EVL  1280
      DO 320 L=1,LMAX                                                           EVL  1290
      MPS = NP(IJ)+L-1                                                          EVL  1300
      KI = MP(MPS)                                                              EVL  1310
      XKI = (XI(KI)-XG(I))/DXA                                                  EVL  1320
      YKI = (YI(KI)-YG(J))/DYA                                                  EVL  1330
  320 FV = FV+AB(MPS)*K(A,B,XKI,YKI,XVD,YVD)                                    EVL  1340
C                                                                               EVL  1350
  340 IF (MODL.NE.1) FV = FV+AL(IAL)+AL(IAL+1)*(XV-XG(I+1))+AL(IAL+2)*(YEVL  1360
     1V-YG(J+1))                                                                EVL  1370
      GO TO 620                                                                 EVL  1380
C                                                                               EVL  1390
C     THIS IS FOR XO(IO),YO(JO)) POINTS WHICH ARE IN TWO RECTANGLES,            EVL  1400
C     (I,J) AND (I+1,J).                                                        EVL  1410
C                                                                               EVL  1420
  360 DYA = YG(J+2)-YG(J)                                                       EVL  1430
      YVD = (YV-YG(J))/DYA                                                      EVL  1440
```

```
C                                                                       EVL 1450
      DO 420 IP=1,2                                                     EVL 1460
      FC(IP) = 0.                                                       EVL 1470
      IS = I+IP-1                                                       EVL 1480
      IJ = (J-1)*NXG+IS                                                 EVL 1490
      IAL = 3*IJ-2                                                      EVL 1500
      IF (MODL.EQ.2) GO TO 400                                          EVL 1510
      DXA = XG(IS+2)-XG(IS)                                             EVL 1520
      XVD = (XV-XG(IS))/DXA                                             EVL 1530
      LMAX = NP(IJ+1)-NP(IJ)                                            EVL 1540
C                                                                       EVL 1550
      DO 380 L=1,LMAX                                                   EVL 1560
      MPS = NP(IJ)+L-1                                                  EVL 1570
      KI = MP(MPS)                                                      EVL 1580
      XKI = (XI(KI)-XG(IS))/DXA                                         EVL 1590
      YKI = (YI(KI)-YG(J))/DYA                                          EVL 1600
  380 FC(IP) = FC(IP)+AB(MPS)*K(A,B,XKI,YKI,XVD,YVD)                    EVL 1610
C                                                                       EVL 1620
  400 IF (MODL.NE.1) FC(IP)=FC(IP)+AL(IAL)+AL(IAL+1)*(XV-XG(IS+1))+AL(IAEVL 1630
     1L+2)*(YV-YG(J+1))                                                 EVL 1640
      A = 0.                                                            EVL 1650
  420 CONTINUE                                                          EVL 1660
C                                                                       EVL 1670
      WI = H5((XV-XG(I+1))/DX)                                          EVL 1680
      FV = FC(1)*WI+(1.-WI)*FC(2)                                       EVL 1690
      GO TO 620                                                         EVL 1700
C                                                                       EVL 1710
C     THIS IS FOR (XO(IO),YO(JO)) POINTS WHICH ARE IN TWO RECTANGLES,   EVL 1720
C     (I,J) AND (I,J+1).                                                EVL 1730
C                                                                       EVL 1740
  440 DXA = XG(I+2)-XG(I)                                               EVL 1750
      XVD = (XV-XG(I))/DXA                                              EVL 1760
C                                                                       EVL 1770
      DO 500 JP=1,2                                                     EVL 1780
      FC(JP) = 0.                                                       EVL 1790
      JS = J+JP-1                                                       EVL 1800
      IJ = (JS-1)*NXG+I                                                 EVL 1810
      IAL = 3*IJ-2                                                      EVL 1820
      IF (MODL.EQ.2) GO TO 480                                          EVL 1830
      DYA = YG(JS+2)-YG(JS)                                             EVL 1840
      YVD = (YV-YG(JS))/DYA                                             EVL 1850
      LMAX = NP(IJ+1)-NP(IJ)                                            EVL 1860
C                                                                       EVL 1870
      DO 460 L=1,LMAX                                                   EVL 1880
      MPS = NP(IJ)+L-1                                                  EVL 1890
      KJ = MP(MPS)                                                      EVL 1900
      XKJ = (XI(KJ)-XG(I))/DXA                                          EVL 1910
      YKJ = (YI(KJ)-YG(JS))/DYA                                         EVL 1920
```

```
  460 FC(JP) = FC(JP)+AB(MPS)*K(A,B,XKJ,YKJ,XVD,YVD)                        EVL 1930
C                                                                          EVL 1940
  480 IF (MODL.NE.1) FC(JP)=FC(JP)+AL(IAL)+AL(IAL+1)*(XV-XG(I+1))+AL(IALEVL 1950
     1+2)*(YV-YG(JS+1))                                                    EVL 1960
      B = 0.                                                               EVL 1970
  500 CONTINUE                                                             EVL 1980
C                                                                          EVL 1990
      UJ = H5((YV-YG(J+1))/DY)                                             EVL 2000
      FV = FC(1)*UJ+(1.-UJ)*FC(2)                                          EVL 2010
      GO TO 620                                                            EVL 2020
C                                                                          EVL 2030
C     THIS IS FOR (XO(IO),YO(JO)) POINTS WHICH ARE IN FOUR RECTANGLES,     EVL 2040
C     (I,J), (I+1,J), (I,J+1), AND (I+1,J+1).                              EVL 2050
C                                                                          EVL 2060
  520 KFC = 0                                                              EVL 2070
C                                                                          EVL 2080
      DO 600 JP=1,2                                                        EVL 2090
      JS = J+JP-1                                                          EVL 2100
      DYA = YG(JS+2)-YG(JS)                                                EVL 2110
      YVD = (YV-YG(JS))/DYA                                                EVL 2120
      A = 1.                                                               EVL 2130
C                                                                          EVL 2140
      DO 580 IP=1,2                                                        EVL 2150
      IS = I+IP-1                                                          EVL 2160
      IJ = (JS-1)*NXG+IS                                                   EVL 2170
      IAL = 3*IJ-2                                                         EVL 2180
      KFC = KFC+1                                                          EVL 2190
      FC(KFC) = 0.                                                         EVL 2200
      IF (MODL.EQ.2) GO TO 560                                             EVL 2210
      IF (IS.GT.1) A = 0.                                                  EVL 2220
      DXA = XG(IS+2)-XG(IS)                                                EVL 2230
      XVD = (XV-XG(IS))/DXA                                                EVL 2240
      LMAX = NP(IJ+1)-NP(IJ)                                               EVL 2250
C                                                                          EVL 2260
      DO 540 L=1,LMAX                                                      EVL 2270
      MPS = NP(IJ)+L-1                                                     EVL 2280
      KI = MP(MPS)                                                         EVL 2290
      XKI = (XI(KI)-XG(IS))/DXA                                            EVL 2300
      YKI = (YI(KI)-YG(JS))/DYA                                            EVL 2310
  540 FC(KFC) = FC(KFC)+AB(MPS)*K(A,B,XKI,YKI,XVD,YVD)                     EVL 2320
C                                                                          EVL 2330
  560 IF (MODL.NE.1) FC(KFC)=FC(KFC)+AL(IAL)+AL(IAL+1)*(XV-XG(IS+1))+AL(EVL 2340
     1IAL+2)*(YV-YG(JS+1))                                                 EVL 2350
  580 CONTINUE                                                             EVL 2360
C                                                                          EVL 2370
      B = 0.                                                               EVL 2380
  600 CONTINUE                                                             EVL 2390
C                                                                          EVL 2400
```

```
      WI = H5((XV-XG(I+1))/DX)                                                          EVL 2410
      UJ = H5((YV-YG(J+1))/DY)                                                          EVL 2420
      FV = WI*(UJ*FC(1)+(1.-UJ)*FC(3))+(1.-WI)*(UJ*FC(2)+(1.-UJ)*FC(4))                 EVL 2430
  620 FO(IO,JO) = FV                                                                    EVL 2440
C                                                                                       EVL 2450
  640 CONTINUE                                                                          EVL 2460
C                                                                                       EVL 2470
      RETURN                                                                            EVL 2480
      END                                                                               EVL 2490
```

```
      FUNCTION K (A,B,U,V,S,T)                                                  K      10
C                                                                               K      20
CCC   THIS FUNCTION EVALUATES THE REPRESENTER FOR THE POINT EVALLATION          K      30
CCC   (AT (U,V)) FUNCTIONAL FOR THE SARD CORNER SPACE B CORNER 2,2.             K      40
CCC                                                                             K      50
CCC   THE ARGUMENTS ARE AS FOLLOWS.                                            K      60
CCC                                                                             K      70
CCC      A,B - INPUT.  THE BASE POINT (A,B) FOR THE SARD SPACE B                K      80
CCC                    CORNER 2,2.                                              K      90
CCC      U,V - INPUT.  THE LINEAR FUNCTIONAL IS EVALUATION AT (U,V).            K     100
CCC      S,T - INPUT.  THE REPRESENTER IS EVALUATED AT (S,T).                   K     110
CCC                                                                             K     120
CCC   BECAUSE OF THE SHORT WORD LENGTH OF THE IBM 360/370 COMPUTERS             K     130
CCC   CERTAIN VARIABLES ARE DECLARED AS DOUBLE PRECISION.  THIS STATE-          K     140
CCC   MENT MAY BE SAFELY REPLACED WITH THE STATEMENT  ' REAL K ' WHEN           K     150
CCC   THIS PROGRAM IS USED ON COMPUTERS WITH LONGER WORD LENGTHS.               K     160
C                                                                               K     170
      DOUBLE PRECISION K,SMA,UMA,USMA,X,TRMS,GP1,GP2                            K     180
      SS = S                                                                    K     190
      UU = U                                                                    K     200
      AA = A                                                                    K     210
      KVAR = 1                                                                  K     220
  100 SMA = SS-AA                                                               K     230
      UMA = UU-AA                                                               K     240
      USMA = SMA*UMA                                                            K     250
      TRMS = 0.                                                                 K     260
      IF (USMA.LE.0.) GO TO 160                                                 K     270
      IF (UMA.GE.0.) GO TO 120                                                  K     280
      UMA = -UMA                                                                K     290
      SMA = -SMA                                                                K     300
  120 IF (SMA.LE.UMA) GO TO 140                                                 K     310
      X = SMA                                                                   K     320
      SMA = UMA                                                                 K     330
      UMA = X                                                                   K     340
  140 TRMS = SMA/2.*(USMA-SMA**2/3.)                                            K     350
  160 GP2 = 1.+USMA+TRMS                                                        K     360
      GO TO (180,200), KVAR                                                     K     370
  180 GP1 = GP2                                                                 K     380
      SS = T                                                                    K     390
      UU = V                                                                    K     400
      AA = B                                                                    K     410
      KVAR = 2                                                                  K     420
      GO TO 100                                                                 K     430
  200 K = GP1*GP2                                                               K     440
      RETURN                                                                    K     450
      END                                                                       K     460
```

```
      DIMENSION X(25), Y(25), F(25), XO(3), YO(3), FO(3,3), FAC(3,3,4),  TST00010
     1 E(3,3), IWK(100), WK(120)                                         TST00020
      DATA NOUT/6/                                                       TST00030
      DATA FAC / 2.5575399 , 1.1562605 , 0.3436082 ,                     TST00040
     1           1.6866026 , 0.7295436 , 0.2116178 ,                     TST00050
     2           0.7374883 , 0.3435128 , 0.0937047 ,                     TST00060
     3           2.6586781 , 1.5227375 , 0.6821580 ,                     TST00070
     4           1.7544727 , 0.8773775 , 0.2846120 ,                     TST00080
     5           0.9337589 , 0.4132084 , 0.1096575 ,                     TST00090
     6           2.5726633 , 1.1580372 , 0.3479553 ,                     TST00100
     7           1.6950045 , 0.7241199 , 0.2139754 ,                     TST00110
     8           0.7366993 , 0.3412316 , 0.0933840 ,                     TST00120
     9           2.3763390 , 1.1071606 , 0.3481252 ,                     TST00130
     A           1.6266966 , 0.7474328 , 0.2177160 ,                     TST00140
     B           0.7119821 , 0.3248314 , 0.0977760 /                     TST00150
      K = 0                                                              TST00160
C                                                                        TST00170
      DO 100 I=1,5                                                       TST00180
      DO 100 J=1,5                                                       TST00190
      K = K+1                                                            TST00200
      X(K) = I-1+1./FLOAT(K)                                             TST00210
      Y(K) = J-1./FLOAT(K)                                               TST00220
  100 F(K) = 4.*EXP(-(X(K)**2+Y(K)**2)*.2)                               TST00230
C                                                                        TST00240
      NP = 25                                                            TST00250
C                                                                        TST00260
      DO 120 I=1,3                                                       TST00270
      XO(I) = I+.5                                                       TST00280
  120 YO(I) = I-.5                                                       TST00290
C                                                                        TST00300
      DO 160 MODE=1,3                                                    TST00310
      NWK = 120                                                          TST00320
      NIWK = 100                                                         TST00330
      CALL LOE22 (MODE,6,NP,X,Y,F,3,XO,3,YO,IWK,NIWK,WK,NWK,FO,KER)      TST00340
C                                                                        TST00350
      DO 140 I=1,3                                                       TST00360
      DO 140 J=1,3                                                       TST00370
  140 E(I,J) = FAC(I,J,MODE)-FO(I,J)                                     TST00380
C                                                                        TST00390
      WRITE (NOUT,2) MODE,KER,NIWK,NWK                                   TST00400
  160 WRITE (NOUT,1) FO,E                                                TST00410
C                                                                        TST00420
      IWK(1) = 2                                                         TST00430
      IWK(2) = 3                                                         TST00440
      WK(1) = .2                                                         TST00450
      WK(2) = 1.5                                                        TST00460
      WK(3) = 3.5                                                        TST00470
      WK(4) = 4.+1./21.                                                  TST00480
```

Deleted

```
THE VALUES OF MODE, KER, NIWK, AND NWK ARE     1     0    79    77

      FUNCTION VALUES
           2.557540              1.156260              0.343608
           1.686603              0.729544              0.211618
           0.737488              0.343513              0.093705

      DEVIATIONS (THESE VALUES REPRESENT ROUNDOFF ERROR)
          -0.000015              0.000042              0.000052
          -0.000009              0.000020              0.000050
           0.000009              0.000038              0.000009

THE VALUES OF MODE, KER, NIWK, AND NWK ARE     2     0    79    37

      FUNCTION VALUES
           2.658678              1.522738              0.682158
           1.754473              0.877377              0.284612
           0.933759              0.413208              0.109657

      DEVIATIONS (THESE VALUES REPRESENT ROUNDOFF ERROR)
           0.000003              0.000002             -0.000001
           0.000001              0.000000             -0.000000
           0.000001             -0.000000              0.0

THE VALUES OF MODE, KER, NIWK, AND NWK ARE     3     0    79   104

      FUNCTION VALUES
           2.572663              1.158027              0.347955
           1.695004              0.724120              0.213975
           0.736699              0.341232              0.093384

      DEVIATIONS (THESE VALUES REPRESENT ROUNDOFF ERROR)
          -0.000018              0.000011              0.000047
          -0.000015              0.000003              0.000045
          -0.000008              0.000040              0.000058

THE VALUES OF MODE, KER, NIWK, AND NWK ARE     3     0    58    76

      FUNCTION VALUES
           2.376339              1.107161              0.348125
           1.626697              0.747433              0.217716
           0.711982              0.324831              0.097776

      DEVIATIONS (THESE VALUES REPRESENT ROUNDOFF ERROR)
           0.000092             -0.000019             -0.000096
           0.000033             -0.000010             -0.000066
           0.000026              0.000023              0.000017
```

REFERENCES

1. Hiroshi Akima, "A Method of Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Points", to appear in Transactions on Math Software.

2. R. E. Barnhill, "Representation and Approximation of Surfaces", in Mathematical Software III, pp. 69-120, John R. Rice, ed., Academic Press, 1977.

3. R. E. Barnhill and J. A. Gregory, "Polynomial Interpolation to Boundary Data on Triangles", Math Comp 29 (1975) 726-735.

4. R. Franke, "Locally Determined Smooth Interpolation at Irregularly Spaced Points in Several Variables", JIMA 19 (1977) 471-482.

5. R. Franke, "On the Computation of Optimal Approximations in Sard Corner Spaces $B_{\lceil p,q \rceil}$ ", Technical Report # NPS- 53Fe76121, Naval Postgraduate School, Monterey, CA (to appear in SIAM J. Numer Anal.).

6. R. Franke, "Reproducing Kernel Functions in Sard Corner Spaces", Technical Report # NPR-53-78-001, Naval Postgraduate School, Monterey, CA.

7. C. L. Lawson, "Software for C' Surface Interpolation", in Mathematical Software III, pp. 161-193, John R. Rice, ed., Academic Press, 1977.

8. L. L. Schumaker, "Fitting Surfaces to Scattered Data", in Approximation Theory II, pp. 203-268, edited by G. G. Lorentz, C. K. Chin, and L. L. Schumaker, Academic Press, 1976.

DISTRIBUTION LIST

Defense Documentation Center Cameron Station Alexandria, VA 22314	2
Dudley Knox Library Naval Postgraduate School Monterey, CA 93940	2
Dean of Research Naval Postgraduate School Monterey, CA 93940	2
Department of Mathematics C. O. Wilde, Chairman Professor C. Comstock Professor F. Faulkner Professor A. Schoenstadt Professor R. Franke Naval Postgraduate School Monterey, CA 93940	 1 1 1 1 5
Dr. Richard Lau Office of Naval Research 1030 East Green St. Pasadena, CA 91106	1
Professor R. E. Barnhill Department of Mathematics University of Utah Salt Lake City, UT 84112	1
Professor G. M. Nielson Department of Mathematics Arizona State University Tempe, AZ 85281	1
Chief of Naval Research ATT: Mathematics Program Arlington, VA 22217	2
Rosemary E. Chang Sandia Laboratories Applied Mathematics, Division 2-8235 Livermore, CA 94550	1
Hiroshi Akima Office of Telecommunications Department of Commerce Boulder, CO 80302	1

Professor J. A. Bishop 1
Department of Genetics
University of Liverpool
P. O. Box 147
Liverpool L693BX, GREAT BRITAIN

Dr. Bill Gordon 1
Office of Naval Research/London
Box 39
FPO New York, NY 09510

W. J. Dejka 1
Naval Ocean Systems Center
San Diego, CA 92152

Professor R. Wait 1
Department of Computational and Statistical Science
University of Liverpool
P. O. Box 147
Liverpool, L693BX, GREAT BRITAIN

Dr. Herbert Steinberg 1
Mathematical Applications Group, Inc.
3 Westchester Plaza
Elmsford, NY 10523

Dr. D. Sastry 1
Department of Aeronautical Engineering
Indian Institute of Science
Bangalore - 560012, INDIA

Mark Raterman 1
Silesian University, Institute of Chemistry
9, Szkolna Street, 40-006
Katornice, POLAND

Dr. C. L. Lawson 1
Information Systems Division
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91103

Computer Center
Professor D. G. Williams 1
Mr. Roger Hilleary 1
Naval Postgraduate School
Monterey, CA 93940

International Mathematical and Statistical Libraries, Inc. 1
7500 Bellaire Blvd.
Houston, TX 77036

U182403

DUDLEY KNOX LIBRARY - RESEARCH REPORTS
5 6853 01060594 2